Doctoral theses at NTNU, 2024:215

Daniel Tianhou Zhang

Asynchronous and Infinite Replica Exchange Transition Interface Sampling

Doctoral thesis

NTNU
Norwegian University of Science and Technology
Thesis for the Degree of
Philosophiae Doctor
Faculty of Natural Sciences
Department of Chemistry

NTNU
Norwegian University of
Science and Technology

Daniel Tianhou Zhang

Asynchronous and Infinite Replica Exchange Transition Interface Sampling

Thesis for the Degree of Philosophiae Doctor

Trondheim, May 2024

Norwegian University of Science and Technology
Faculty of Natural Sciences
Department of Chemistry

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the Degree of Philosophiae Doctor

Faculty of Natural Sciences
Department of Chemistry

ISBN 978-82-326-8022-1 (printed ver.)
ISBN 978-82-326-8021-4 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

Doctoral theses at NTNU, 2024:215

Printed by NTNU Grafisk senter

ABSTRACT

Simply by pressing a key to run Molecular Dynamics (MD) simulations, the dynamics of complex molecular systems play out as if seen by a powerful microscope. From modeling chemical reactions to nucleation and protein folding, MD is therefore a valuable tool that utilizes computer calculations to predict useful, real-world information. Predicting certain properties like the reaction rate constant, however, proves to be exceedingly difficult, as standard MD simulations rarely undergo spontaneous reactions by themselves and often remain within one stable state for insurmountable amounts of time instead.

One way to sample rare events more quickly could be to use the Monte Carlo (MC) based Replica Exchange Transition Interface Sampling (RETIS) method. By defining specific start and stop conditions, MC moves like *shooting* are employed to efficiently generate statistically rare and unbiased MD trajectories. However, due to also employing replica exchange moves, RETIS is most efficiently executed sequentially. As a result, simulations may still take several months or even over a year to converge. While significantly better than straightforward MD, the long simulation times hinder widespread applicability. This thesis, therefore, presents major enhancements to accelerate the RETIS convergence rate. Particularly, we enhance RETIS with a new, highly decorrelative *shooting*-based move called Wire Fencing, minimize rejection by applying the high-acceptance procedure, enable the calculation of infinite replica exchange swaps without the factorial cost, and derive a parallel RE scheme with linear MD scaling. Connecting these enhancements together forms the ∞RETIS protocol, which enables the calculation of rate constants in days or weeks instead. Finally, we apply the implemented ∞RETIS software *infretis* to study the electron transfer reaction between two ruthenium ions and the formation of carbonic acid from solvated carbon dioxide.

ACKNOWLEDGEMENTS

In the span of 3 years and 9 months, the COVID-19 pandemic came and went, machine learning models invaded my personal life (art) and I appear to have written a PhD thesis.

Sketch using a fountain pen and a brush pen.

Honestly, I am a pretty tsundere[1] when it comes to compliments and acknowledging people, but I will try my best. First and foremost, I would like to thank prof. Titus van Erp for being my supervisor. From attending your thermodynamics and molecular modeling lectures pre-PhD to the fruitful discussions we've had online and in-person, this PhD journey from start to completion has been really enjoyable and would not have been possible without you. I would like to thank prof. Enrico Riccardi for the early guidance in my PhD study, for joining the many pair programming sessions we have done together, and for inviting me and Titus to the University of Stavanger. I want to thank prof. Henrik Koch and prof. Ida-Marie Høyvik for also being my co-supervisors. While our collaborative efforts on the excited state dynamics project did not yield sufficient progress to be included in my thesis, I am still optimistic about completing and publishing the electron transfer study in the ruthenium ion system, as discussed in Chapter 4. For that work, I am also grateful for the work put in by Benjamin and the guidance given by prof. Bernd Ensing. I want to thank prof. An Ghysels and PhD

[1] a japanese term for an outwardly cold character that is not good at giving compliments (among other things).

student Wouter for our rewarding collaborations together, and also for allowing me to visit Ghent University for a week. The (tilted) maze potential was quite fun to set up and run.

As one of the few path samplers in the world, I want to thank dr. Sander Roet for being my path sampling mentor and my go-to rubber ducky, even after you defended your thesis. Our walk through Amsterdam was very nice. As a neighbor both at work and off work, I want to thank prof. Anders Lervik for being a good neighbor, lending me tea and for the helpful discussions we have had. I would like to thank Titus, Sander, Tor, Lukas, Hilde and Fiona for proofreading this thesis. Apparently (and luckily), explaining concepts through analogies like panda and human dynamics seems to be appreciated. I would also thank my office mates, Regina and Lukas, for being very nice office mates.

I want to thank Sander, Regina, Alex, Linda, Inge, Per-Olof, Anders, Andreas, Tor, Rosario, Lukas, Marcus, Eirik, Sarai, Bendik, Sara, Matteo, Federico, Yassir, Alice, Jan Haakon, Jacob, Konrad, Benjamin, Jan Gustav, Ylva, Wanjing, Lu Xia (and any other people I might have missed) for being amazing colleagues and friends at IKJ. Working here with you guys has been lots of fun. I want to thank my bros who also seemed to find themselves in a PhD position at the same time as me, Martin, Yanzhe, Kiet, Lodin, Sebastian and Dat. Having you guys around to let of steam and to talk with has been assuring.

For showing me how fun research is during my MSc, and for partly determining my choice in starting this PhD, I would like to thank prof. Øivind Wilhelmsen and dr. Vilde Bråten. For inviting me to visit Tokyo Woman's Christian University, and showing me the cool restaurants nearby, I would like to thank prof. Tatsuya Joutsuka and prof. Koji Ando. For all our chit-chat over internet, and for our hiking trips during my holidays, I would thank Michael, Anders, Conrad and Vidar. For letting me stay at your place prior to the TRAINS conference, I want to thank Truc and Timmy. For joining my personal trip to Canada and Japan in 2023, I want to thank Jay, Alice, Tony, Kevin and Yuri. For inviting me to present at your research groups, I want to thank prof. Senbo Xiao and prof. Jianying He. For attending the book club meetings at the various nice cafés in town, I want to thank Lodin, Sebastian, Snorre, Emma, Eirik and Sofie. While being established at the end of my PhD, I still managed to attend quite a few of the Japanese language cafes that we initiated, so I want to thank Tor, Sondre, Jørgen and Nils.

I was initially planning to write a quite short acknowledgment chapter. However, I became convinced to write a more lengthy end explicit version (this version) after a lunch break talk with Eirik, who felt he wrote a too-short acknowledgment chapter in his thesis and seemingly regretted it. I might have regretted it too, so I am grateful for that conversation I had with Eirik that day.

I want to thank Titus for approving the working title of my thesis, "Calculating rare constants took too long, so I accelerated RETIS convergence through wire fencing, parallelization and infinite swapping", and for convincing me out of it.

While I would classify myself as an introvert, I've apparently met and talked to many people throughout this PhD. But none of this could have happened without my family, so thanks for all the support throughout the years.

The latex template used for this thesis comes from https://github.com/ninasalvesen/thesis_latex_template.

CONTENTS

CHAPTER

ONE

INTRODUCTION

Monte Carlo (MD) and Molecular Dynamics (MD) are two important simulation techniques that allow for the modeling of physical systems at the molecular level, such that information like temperature, pressure, energies, molecular geometries and rates can be estimated by running simulations using a computer [1, 2]. While both MC and MD-based methods are being heavily utilized today, they are different classes of methods and have their own strengths and weaknesses. MC deals with using random numbers to obtain numerical results. While the name "The Monte Carlo Method" was popularized by the seminal paper in 1947 of the same name [3], random, stochastic methods have been in use far earlier, like for the estimation of π using random numbers in the early 1800s [4]. The use of random numbers to solve problems has only increased since then, where running MC-based simulations is now an effective way to explore the vast configuration space spanned by molecular systems. MD on the other hand solves Newton's equations of motion numerically to propagate the studied molecular system forward in time. Many of the numerical solutions for these equations predate the existence of computers, like in 1687, where the primary application was to understand the motions of objects in outer space [2]. One of the first applications of these equations to molecular systems was the study of hard spheres [5], a type of particle sibling to the ideal gas particle, but with volume. The simulation of hard-sphere particles using MD and MC is still of high-interest today and has been studied recently for the development of a new thermodynamic state equation for small systems [6]. Considerably more sophisticated molecular systems can also be simulated, however, like the recent MD simulations of the SARS-CoV-2 spike protein that were run to generate mechanistic understanding for the development of vaccines and antiviral agents [7]. Depending on the studied molecular system and the sought information, however, MC or MD simulations might have to be tailored in some way in order to generate sufficiently accurate data that can be comparable to experimental results. For example, if high statistical errors are present, then running the simulation for a longer time could possibly solve the issue. Or if the system must be studied at a certain temperature or pressure, then a thermostat or a barostat should be applied. Or for simulations that require molecular bonds to be broken, such as the deprotonation of hydrofluoric acid [8],

$$\mathrm{HF_{(aq)}} + \mathrm{H_2O_{(aq)}} \rightarrow \mathrm{F^-_{(aq)}} + \mathrm{H_3O^+_{(aq)}} \tag{1.1}$$

then the desired bond-breaking event can only occur if MC or MD is run with molecular force fields that include such occurrences, like ReaxFF [9, 10], or by calculating forces from "the ground up" using more expensive *ab initio*-based methods like Density Functional Theory [11]. Both of these molecular models can be expensive in terms of computational power [12], however, and may demand excessively more time to converge calculated results compared to simulations running classical molecular force fields, which are simple parameterized functions describing the forces between atoms and molecules [13]. The drawback of classical molecular force field simulations is that they cannot model bond breaking.

In addition to molecular simulations only being able to model a finite number of particles with high enough accuracy, a central property to be maximized is the simulation efficiency. In the context of computer simulations, efficiency is the time or computational power that is required to obtain results within a certain range of accuracy, so an increase in efficiency means that equivalent results can be generated by using less time or computational power. Historically, simulation efficiency has continuously been increasing through the incessant hardware, software, and algorithmic improvements over the years. One particular way to increase the efficiency of MC simulations, for example, has been through the algorithmic development of replica exchange (RE) moves, also called parallel tempering [14]. The idea of RE is to improve the sampling quality of MC simulations by running several MC simulations in parallel, with each simulation sampling its own distinct distribution through varying a simulation variable like the temperature. RE moves are then performed at certain points during this parallel simulation, swapping the sampled molecular configurations between the simulations. This enables configurations generated by high-temperature ensembles, that can more easily traverse energy barriers, to be sampled by lower-temperature simulations. Without RE, simulating only one low-temperature ensemble to sample configurations separated by energetic barriers could prove to be exceedingly difficult. The introduction of RE moves to MC simulations may not be straightforward, however, as RE also introduces simulation parameters that can affect the simulation efficiency. Examples include the frequency of performing RE moves and how large the temperature differences should be between the parallel simulations [15]. Recent focus has been on finding the optimal parameters that would maximize REMC efficiency, like taking the RE limit of performing a (theoretical) infinite number of RE moves between other types of MC moves [16, 17]. Other types of MC moves to run alongside RE include running MD for a certain amount of time to obtain a new configuration (also called Hybrid Monte Carlo) [18]. In terms of hardware and software improvements, MD has particularly appreciated the recent advancements to graphics processing units (GPU) pushed by the gaming and machine learning communities [19]. These days, MD and MC simulations are often performed on high-performance computing (HPC) clusters consisting of hundreds of nodes, with many of them being equipped with several GPUs [20].

The work included in this thesis revolves around developing methods that solve the *rare event problem*, which is often present when running MD simulations. For example, if one were to study the pyramidal/umbrella inversion reaction of solvated phosphine [21] using MD,

$$\mathrm{H_3P_{(sol)}} \rightarrow \mathrm{PH_{3(sol)}} \tag{1.2}$$

then the presence of an energy barrier (Figure 1.0.1 **Left**) would minimize the umbrella inversion occurrences, generating perhaps zero inversions even if a reasonably long simulation has been run. Additionally, compared to experiments, the enforced small time step and simulated system size exacerbate the sampling rarity due to numerical stability constraints and hardware limitations. This kind of sampling problem is general and can affect any type of molecular system that exhibits regions of local configuration minima separated by an energetic barrier. Typical solutions to this problem revolve around running MC that employ unphysical moves, or biasing the system with additional force through running enhanced sampling methods like umbrella sampling [22] or metadynamics [23]. While yielding statistically correct data, none of the methods generate true dynamical trajectories of the reaction, which can be important to understand initiation conditions and for the calculation of rate constants. A way to circumvent bias while also allowing for the effective sampling of dynamical rare events could be to employ so-called *path sampling* simulations, like Transition Path Sampling (TPS) [24, 25], Transition Interface Sampling (TIS) [26, 27], or Replica Exchange Transition Interface Sampling (RETIS) [28, 29, 30]. Building on the idea of MC schemes sampling MD trajectories instead of configurations [31], path sampling simulations are a class of MC methods that sample statistically rare MD trajectories as if they were cut from a long MD simulation (see Figure 1.0.1 **Right**).

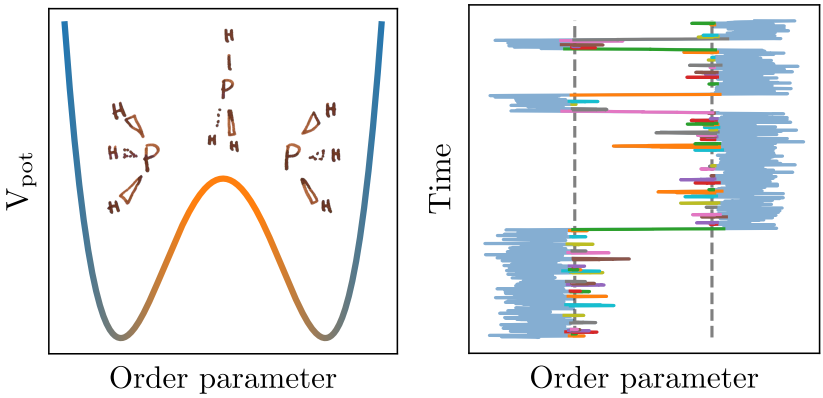


Figure 1.0.1: Left: A double-well potential modeling the umbrella inversion of phosphine. Compared to equilibrium (standard) MD that mainly explore configurations at the two potential minima, path simulations utilize unbiased MD to exclusively explore the more orange-based region. **Right:** The combined pale-blue and colored trajectory shows the time evolution of the order parameter from an equilibrium MD run. By separating stable states via specific order parameter values (horizontal dotted lines), a path simulation (not necessarily in the form of TPS or RETIS) would sample only the colored parts, reducing the total number of force calculations in this example by 90%.

By defining an order parameter that enables the characterization of stable reactant and product states, path simulations force the simulated system to always propagate in between stable states by starting and stopping MD at certain conditions. An essential benefit provided by path sampling methods is that no understanding of the transition mechanism or the transition state is required before running the simulations. Rather, those properties are what one may obtain from analyzing the data produced by the path simulations. While the experimentally comparable rate constant k_{AB} can be calculated through TPS, to do so can be cumbersome as TPS only samples reactive trajectories. Instead, a more effective path sampling method to calculate the rate was developed in the form of TIS, followed by the more efficient RETIS method. Compared to TPS, TIS and RETIS can efficiently calculate the rate constant through running a series of path sampling simulations, each sampling their own distribution of rare, but not necessarily reactive MD trajectories. Other flavors of path sampling simulations include multiple state TPS [32] that directly sample the transitions between multiple stable states, and partial path TIS/RETIS that enable efficient sampling of diffusive barriers [33, 34].

For the work presented in this thesis, I mainly focus on the RETIS method. While exponentially more efficient than MD, RETIS simulations can still take months to a year to converge [Paper C, Chapter 4.1]. A major objective of the papers in this thesis has therefore been to accelerate RETIS convergence through algorithmic improvements, like the implementation of the decorrelating subtrajectory MC move called Wire Fencing (paper B [35]), enabling the calculation of infinite replica exchange, and asynchronous parallelization of RETIS (papers A [36] and C [37]). As a side effect, I have also been involved in the development of alternative path sampling schemes (paper II [34]) and the development of the PyRETIS 3 software (paper III [38]). I also finalized an unrelated paper based on my master's thesis (paper I [6]). Of course, path sampling algorithms are only useful when applied, so another part of my PhD has been to study the redox reaction between solvated Ruthenium ions (Chapter 4.1) and the formation of carbonic acid from carbon dioxide (Chapter 4.2).

In this chapter, I presented a short overview of the various methods to simulate chemical systems at the molecular scale and introduced the rare event problem, which is what the major algorithms TPS, TIS and RETIS have been designed to solve. Chapter 2 of this thesis introduces the theoretical background for the main tools utilized for my research: MD, MC, TPS and RETIS. These methods have thoroughly been described in several textbooks [1, 2, 13, 39] and publications [24, 25, 26, 27, 29, 40, 41], so for this chapter only, the theoretical background will be introduced through self-concocted analogies that I found useful when explaining my research to students or non-specialists. Chapter 3 discusses the contents of papers A, B and C, namely the algorithmic enhancements developed for RETIS. Chapter 4 provides a preliminary discussion on the application of the ∞RETIS protocol to two unpublished applications, and Chapter 5 provides concluding remarks and future outlooks.

Publications included in this thesis

Paper A:

Exchanging Replicas with Unequal Cost, Infinitely and Permanently
Sander Roet, **Daniel Tianhou Zhang**, Titus Sebastiaan van Erp
J. Phys. Chem. A 126, 8878–8886 (**2022**)

Paper B:

Enhanced path sampling using subtrajectory Monte Carlo moves
Daniel Tianhou Zhang, Enrico Riccardi, Titus Sebastiaan van Erp
J. Chem. Phys. 158, 024113 (**2023**).

Paper C:

Highly Parallelizable Path Sampling with Minimal Rejections Using Asynchronous Replica Exchange and Infinite Swaps
Daniel Tianhou Zhang; Lukas Baldauf, Anders Lervik, Sander Roet, Titus Sebastiaan van Erp
PNAS 121-7 (**2024**)

Publications in progress

- Electron Transfer Dynamics between Ruthenium Ions in Water (preliminary results discussed in Chapter 4.1)
- Formation of Carbonic Acid from CO_2 using ∞RETIS (preliminary results discussed in Chapter 4.2)

Publications not discussed in this thesis

Paper I:

Equation of state for confined fluids
Vilde Bråten, **Daniel Tianhou Zhang**, Morten Hammer, Ailo Aasen, Sondre Kvalvåg Schnell, Øivind Wilhelmsen
J. Chem. Phys. 156, 244504 (**2022**)

Paper II:

Path sampling with memory reduction and replica exchange to reach long permeation timescales.
Wouter Vervust, **Daniel Tianhou Zhang**; Titus Sebastiaan van Erp, An Ghysels
J. Biophys. 122, 2960–2972 (**2023**)

Paper III:

PyRETIS 3: Conquering Rare and Slow Events Without Boundaries
Wouter Vervust, **Daniel Tianhou Zhang**, Titus Sebastiaan van Erp, An Ghysels, Enrico Riccardi
J. Comput. Chem 2024, 1–11. **(2024)**

CHAPTER
TWO

THEORETICAL BACKGROUND

This chapter presents topics important for understanding the following chapters 3 and 4, and papers A, B and C. Here, molecular dynamics (MD), replica exchange Monte Carlo (REMC) [40] and the path sampling methods Transition Path Sampling (TPS) [24, 25, 41] and Replica Exchange Transition Interface Sampling (RETIS) [28, 29, 30] are introduced. For more in-depth discussions, I would look into the cited papers and the following books [1, 2, 13, 39].

2.1 Molecular Dynamics

Say we are a big fan of the *three-body problem* book series by Cixin Liu [42], which involves a story about a chaotic solar system containing three interacting stars in space. Given that a humanoid species lives on a planet revolving around this complex system, then they would want to predict the future positions of all the celestial bodies. However, since no general analytic solution for the three-body problem exists [43], they have to settle for approximate solutions involving uncertainties, such as solving Newton's equations of motion numerically,

$$f = m \cdot a \tag{2.1}$$

where the force f equals mass m times acceleration a. To do so using a computer, a dynamics software can be written up to place balls within a 3D box representing the stars and planets in the solar system. By defining some function $u(r)$ that determines the potential energy based on the positions $r = (r_1, r_2, r_3, \dots)$ of the various celestial bodies, then the force in a certain direction x can be calculated as the negative derivative,

$$f_x = -\frac{\partial u\left(r\right)}{\partial x} \tag{2.2}$$

With the addition of some initial velocities v, the equations of motion can be integrated to obtain numerical solutions that predict the positions of the balls r at a certain timestep Δt in the future. With a numerical error of $\mathcal{O}(\Delta t^2)$, one solution is called the velocity-Verlet algorithm [1]:

$$r(t + \Delta t) = r(t) + v(t)\Delta t + \frac{f(r\left(t\right))}{2m}\Delta t^2 \tag{2.3}$$

$$v(t+\Delta t) = v(t) + \frac{f(r\,(t+\Delta t)) + f(r\,(t))}{2m}\Delta t \tag{2.4}$$

By iteratively calculating the forces and updating the positions, frame-by-frame trajectories describing the dynamical time evolution of the star system can be generated. See Figure 2.1.1 for one "snapshot" of the star dynamics,

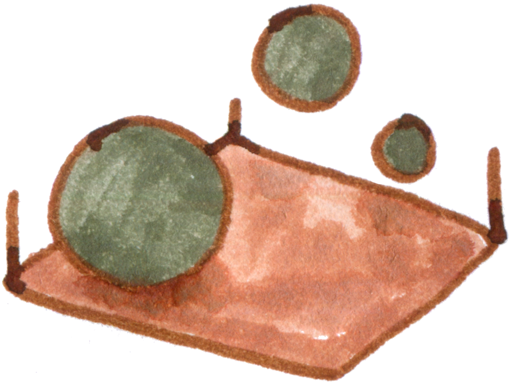


Figure 2.1.1: An illustration of three stars (in green) enclosed within a simulation box.

With this newly developed cosmic simulation software, a humanoid graduate student decides that a smart idea could be to increase the model predictability by simulating the planets at the molecular level. Calling this simulation formalism for Molecular Dynamics (MD), the fine dynamical details down to crystal cell vibrations, chemical reactions, crystal nucleation, liquid-gas evaporation and protein folding can be described in great detail. To do so, the ball masses should be scaled with respect to atomic weights while atomic forces can be modeled using mathematical functions describing bonded, angle and torsion interactions. For example, a bonded interaction can be modeled as a harmonic oscillator,

$$u_{\text{bond}}\ (r) = \frac{a}{2}\left(l(r) - b\right)^2 \tag{2.5}$$

with $l = |r_i - r_j|$ being the bond length between two particles r_i and r_j, and a and b are parameterized coefficients. Other contributions can arise from angle, dihedral, van der Waals and electrostatic interactions [13]. The combined set of functions modelling the behavior of atoms and molecules can together be called molecular force fields, which prove to be accurate for a range of conditions but can be unreliable outside their parameterization. For example, complications can occur when simulation occurs at conditions that favor chemical bond breaking or when molecules cannot be assumed to behave classically, like during photochemical reactions [44, 45]. A way to solve these issues could be to obtain molecular forces from more costly quantum mechanical calculations instead. Solving the Schrodinger equation requires no parameterized data, and various numerical solution schemes exist, such as Density Functional Theory [11] and Coupled Cluster theory [46]. However, compared to force field-based MD that can now simulate 1.6 billion atoms [47], quantum-based MD are generally restricted to much smaller system sizes due to steep compute scaling.

To account for time-demanding calculations and other complexities that molecular systems have to offer, the molecular dynamics formalism can be enhanced with various algorithms and hardware. Apart from previously mentioned MD-related methods and fundamental techniques like periodic boundary conditions and Ewald summation [1, 39], here I mention two recent developments in the MD field:

- Hardware acceleration: The speed of (force field-based) MD simulations continuously increase with the development and demand of hardware, especially in recent years due to GPU acceleration support available in many packages like GROMACS [48] and LAMMPS [49]. Last year (2023) provided us with GPU-accelerated, multi-node support for MD simulations that has scalability for big particle systems, such as a factor of 21x in *ns/day* increase was achieved for a 12 million particle system on a 256-GPU setup compared to legacy code [50]. Here, *ns/day* stands for nanoseconds of simulation time per wall time in days.

- Machine learning force fields: Recent advances in machine learning (ML) extends to the field of molecular modeling, with one application being ML-based force fields (MLFF) [51]. By learning the interactions between atoms from *ab initio* calculations, the dynamics can be simulated at the same accuracy but without the same costs. Consequently, the extra speedup has already opened up the possibility to study various systems that have previously been too expensive [51, 52].

Connecting various algorithms, software, and hardware together, the dynamics of complicated molecular systems can be simulated to predict useful real-world information. For example, the dynamics of boiling water can be modelled [53, 54], as shown in Figure 2.1.2,

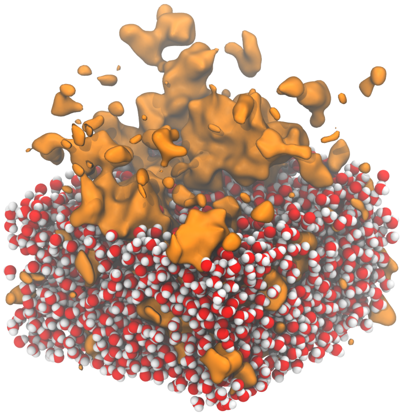


Figure 2.1.2: A MD snapshot of 4055 water molecules simulated at 300 °C and 1 bar. The "bubbles" are illustrated as the yellow blobs, with the water in the upper half of the simulation box hidden. Credit to Anders Lervik for generating this (unused) figure (for Paper C).

2.2 Replica Exchange Monte Carlo

Say one wishes to explore the equilibrium distribution of the body configurations created by an immortal person called Bob living on a never changing planet α. Since he is a human, he will likely spend large amounts of time in named local minima states like *standing*, *sitting*, *lying*, and perhaps very infrequently visit yoga poses like *downward facing dog* and *happy baby* (shown in Figure 2.2.1). One way to sample this configuration distribution could be by running human dynamics (HD), but then the dynamics of Bob have to be followed through time, and depending on the personality of Bob (like if he is an introvert Norwegian), sampling outlier configurations could take large amounts of time.

Figure 2.2.1: Left: The downward facing dog pose. **Right:** The happy baby pose.

If one is indifferent to dynamics, then an alternative method to sample body configurations could be through the use of random numbers. More specifically, one can utilize the well-established Metropolis Monte Carlo algorithm [55] to explore this configuration phase space by letting the configuration state x go to another x' via so-called MC moves. To enforce correct sampling, one could follow detailed balance,

$$\rho(x)P\left(x' \mid x\right)=\rho\left(x'\right)P\left(x \mid x'\right) \tag{2.6}$$

where the statistical weight $\rho(x)$ of state x multiplied by the transition probability of generating state x' from x, $P\left(x' \mid x\right)$, should be equal to the reverse at equilibrium. If the transition probability is now expressed as a product of generation and acceptance probabilities, $P\left(x' \mid x\right)=P_{\text{gen}}\left(x' \mid x\right)P_{\text{acc}}\left(x' \mid x\right)$, then an expression of the acceptance probability can be derived,

$$P_{\text{acc}}\left(x' \mid x\right)=\min\left(1, \frac{\rho\left(x'\right)P_{\text{gen}}\left(x \mid x'\right)}{\rho(x)P_{\text{gen}}\left(x' \mid x\right)}\right) \tag{2.7}$$

Examples of MC moves determining $P_{\text{gen}}\left(x' \mid x\right)$ could be *body-part displacement*, *body-part rotation*, and *running a short HD simulation*, where a general goal in designing MC moves is that the acceptance rule, Equation 2.7, should be as simple as possible to evaluate, which can be done by making the MC move symmetric,

$$P_{\text{gen}}\left(x' \mid x\right)=P_{\text{gen}}\left(x \mid x'\right) \tag{2.8}$$

as symmetry would simplify the equation into only a calculation of weights. Using *body-part displacement* as an example, $P_{\text{gen}}\left(x'|x\right)$ can be formulated as selecting a random body part (BP), then displacing the BP a certain random amount and in a random direction to produce x',

$$P_{\text{gen}}\left(x' \mid x\right) = P_{\text{sel}}\left(\text{BP}|x\right) P_{\text{BPD}}\left(x \to x'|\text{BP}\right) \tag{2.9}$$

Given that Bob remains whole, both the probability of selecting a BP, $P_{\text{sel}}\left(\text{BP}|x\right)$, and the probability of BP displacement $P_{\text{BPD}}\left(x \to x'|\text{BP}\right)$, can be designed to be symmetric and canceled out in its acceptance rule. Other examples of MC moves could be *teleportation*, *time travel*, and *body rotation with respect to some axis*. While performing only the latter set of MC moves does not ensure ergodic sampling, as they generate the same body configuration over and over again, they may significantly boost the sampling of unique configurations in combination with the first set of moves. For example, being teleported to a yoga studio would more likely result in more yoga configurations to be sampled. The α distribution can thus be sampled by repeating the following algorithmic (MC) loop,

1. Perform a MC move to generate a new configuration x' from the old, last accepted configuration x.

2. Accept or reject the new state according to Equation 2.7. If rejected, resample x, else sample x' and set x' as the last accepted state $x \to x'$.

3. Exit loop if convergence has been met. Otherwise, go to step 1.

Averages and errors can thus be calculated from the sampled configurations generated through accepted and rejected MC moves,

$$\{x_0, x_1, x_1, x_2, x_3, x_4, \cdots\} \tag{2.10}$$

Obtaining good, converged statistics within a reasonable time frame can be difficult, however, depending on various factors like the studied system. Even if arriving at a yoga session via a teleportation MC move, committing to perform actual yoga poses can still be difficult due to the introvert Bob nature.

One way to solve this problem could be to introduce the concept of replica exchange MC (REMC) [40], where a set of independent parallel Bob worlds β, γ, δ, ϵ are additionally considered, but compared to the sober world α, the other worlds continually influence their Bobs with factors like alcohol, cocaine, weed and interest in golf. Through series of standard MC moves in each ensemble, the other Bob distributions (ensembles) can also be sampled, which obviously differ from the original α distribution. To enhance sampling, the so-called replica exchange (RE) MC move can also be performed, which simply swaps the last sampled states between two parallel worlds. For example, a RE move between ensembles α and β would yield the general RE acceptance criteria of,

$$P_{\text{acc}}\left(x \in \alpha \leftrightarrow x' \in \beta\right) = \min\left[1, \frac{\rho_\alpha\left(x'\right)\rho_\beta\left(x\right)}{\rho_\alpha\left(x\right)\rho_\beta\left(x'\right)}\right] \tag{2.11}$$

where if accepted, the last accepted states between the two worlds are swapped (and then sampled), allowing access to faster and direct sampling of other, more rare configurations present in both α and β worlds. To maintain detailed balance, the attempt to sample new states must occur concurrently for all the ensembles, usually in the form of performing either a *set of standard MC moves* or a *set of replica exchange moves* such that the number of MC move attempts remains the same for all ensembles. The probability of performing either of the move sets should together sum to one,

$$P_{\mathrm{MC}} + P_{\mathrm{RE}} = 1 \tag{2.12}$$

where P_{RE} is generally set to a value of around 0.5 [16]. For example, if P_{MC} is selected, then a standard MC move must be run in each ensemble α-ϵ. Vice versa, swapping must occur for all ensembles if P_{RE} is chosen. For REMC simulations with ensembles not being able to participate in the swap move (like simulations with an odd number of ensembles), the zero-move can be performed instead (to resample the old state). Even if only one distribution is of interest, like distribution α, the convergence speed up can often be worth the extra computing cost in running multiple ensembles [1]. The following Table 2.2.1 illustrates the concurrent sampling of all ensembles α-ϵ, with cycles of body-part displacement (BPD) moves and RE moves being performed with a probability of 50%,

Table 2.2.1: An REMC scheme illustrating the sampling development of states a-e within ensembles α-ϵ with each cycle having a 50% chance to perform BPD/RE moves. This diagram illustrates BPD acceptance $\mathrm{a}_0 \rightarrow \mathrm{a}_1$ in α, rejection $\mathrm{d}_0 \rightarrow \mathrm{d}_0$ in δ, the null move $\mathrm{b}_1 \rightarrow \mathrm{b}_1$ in α, and RE acceptance between various ensembles.

		BPD		RE		RE		BPD	
α	a_0	→	a_1	→	b_1	→	b_1	→	b_2
β	b_0	→	b_1	→	a_1	→	d_0	→	d_1
γ	c_0	→	c_1	→	d_0	→	a_1	→	a_2
δ	d_0	→	d_0	→	c_1	→	e_0	→	e_1
ϵ	e_0	→	e_0	→	e_0	→	c_1	→	c_1

Of course, the REMC idea is readily extended to other applications like sampling the distributions of molecular systems and can outperform MD in situations with various energy barriers between conformations, such as in the case of umbrella inversion of phosphine [21]. Compared to MD, MC does not sample dynamics and can easily employ non-physical MC moves to move between minima, like randomly displacing one or multiple atoms.

2.3 Path Sampling

Say we're writing a panda dynamics (PD) software to study the local initiation conditions for the occurrence of the panda baby-making process. However, given that the computer model yields realistic panda behaviour, then the generated results will simply be the same as what is observed in real life, that the occurrence of such events can be quite rare. Therefore, even when performing long, brute-force panda dynamics simulations, most of the resulting data would simply be useless and non-reactive in the form of pandas eating, sleeping, and rolling around (see Figure 2.3.1). Additionally, since the dynamical trajectory leading up to the rare event is of interest, applying schemes that only sample configurations, like MC, umbrella sampling [22], or metadynamics [23] will not directly help.

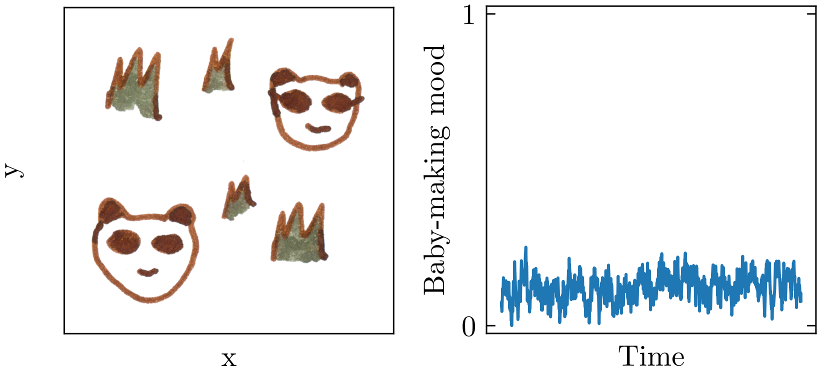


Figure 2.3.1: Left: A panda dynamics snapshot showing two pandas. **Right:** The baby-making mood plotted against time.

2.3.1 Transition Path Sampling

As an alternative, reactive and unbiased PD trajectories can be sampled via the MC-based Transition Path Sampling (TPS) [25] method. To do so, a viable TPS state (path) can be defined to equal a set of consecutive PD phase points $X = \{x_0, x_1, \ldots, x_M\}$ describing a reactive event (the occurrence of a baby making process). Paths can be determined to be reactive or not via an order parameter function $\lambda\,(x)$ that determines the progress of a phase point, like whether the phase point is in state A, $\{x|\lambda(x) < \lambda_A\}$, state B, $\{x|\lambda(x) > \lambda_B\}$, or somewhere in between, $\{x|\lambda_A < \lambda(x) < \lambda_B\}$. In the PD case, the order parameter function can simply be the function that calculates the baby-making mood, with an educated placement of $\lambda_A = 0.2$ and $\lambda_B = 0.99$. Thus, a reactive path can be defined as having the first frame $x_0 \in A$, the last frame $x_0 \in B$, and frames $\{x_1 \ldots x_{M-1}\} \notin (A \cup B)$. Based on these rules, the statistical weight of a trajectory X can equal,

$$\rho_{\rm TPS}(X) = h_{\rm TPS}\,(X)\,\rho\,(x_0) \prod_{i=0}^{M-1} p\,(x_i \to x_{i+1}) = h_{\rm TPS}\,(X)\,\rho\,(x_0)\,P_{\rm PD}(X|x_0) \tag{2.13}$$

where $h_{\rm TPS}\,(X)$ is an indicator function having a value of 0 or 1 depending on whether path X is a valid TPS path or not,

$$h_{\rm TPS}\,(X) = \begin{cases} 1 & \text{if } x_0 \in \text{A and } x_M \in \text{B and } \{x_1 \ldots x_{M-1}\} \notin (\text{A} \cup \text{B}) \\ 0 & \text{otherwise} \end{cases} \tag{2.14}$$

$\rho\,(x_0)$ is the statistical weight of phase point x_i, and $P_{PD}(X|x_0)$ is the product probability of generating path X from phase point x_0 via steps of PD probabilities $p\,(x_i \to x_j)$. Note here that this TPS derivation constitutes the *flexible path* version of TPS, as no constant length constraints are applied to $h_{\rm TPS}\,(X)$. Utilizing the microscopic reversibility condition,

$$\rho\,(x_i)\,p\,(x_i \to x_j) = \rho\,(x_j)\,p\,(\bar{x}_j \to \bar{x}_i) \tag{2.15}$$

where $\bar{x}_i$ is x_i with reversed velocities, allows the product $\rho\,(x_0)\,P_{\rm PD}\,(X \mid x_0)$ in Equation 2.13 to be rearranged to use any phase point x_i of the path,

$$\rho\,(x_0)\,P_{PD}(X|x_0) = \rho\,(x_i)\,P_{PD}(X_b|\bar{x}_i)P_{PD}(X_f|x_i) = \rho\,(x_i)\,P_{PD}(X|x_i) \tag{2.16}$$

with X_b and X_f in this case being the backward and forward part of path X from shooting point x_i. A new path $X^{(n)}$ can be generated from an old path $X^{(o)}$ via the so-called *shooting* MC move, which, excluding the first and last phase points, selects an arbitrary point of the old path $X^{(o)}$, modifies the velocities, and then propagates the new phase point backward and forward in time to hopefully generate a new valid trajectory,

$$P_{\text{gen}}\left(X^{(o)} \to X^{(n)}\right) = P_{\text{sel}}\left(x^{(o)} \mid X^{(o)}\right) P_{\text{vel}}\left(x^{(o)} \to x^{(n)}\right) P_{\text{PD}}\left(X^{(n)} \mid x^{(n)}\right) \quad (2.17)$$

where $x^{(o)}$ and $x^{(n)}$ are phase points that share system positions but differ in velocities, and is somewhere in between the first and last phase points of both paths, see Figure 2.3.2 **left**,

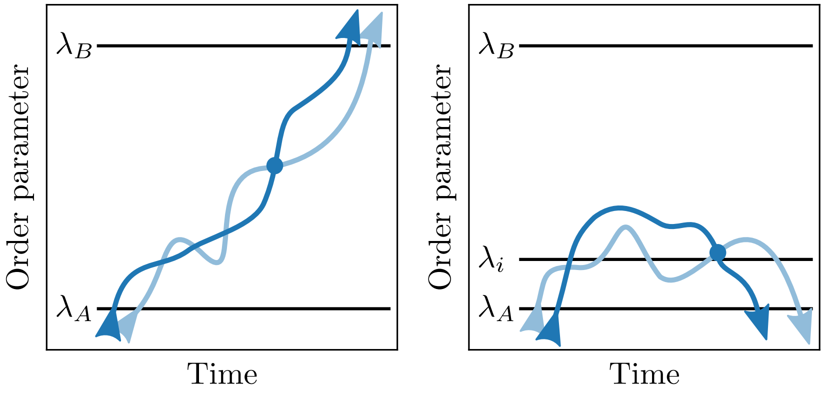


Figure 2.3.2: The shooting move to produce path blue from path light blue in TPS (**Left**) and TIS (**Right**).

Again, to ensure correct sampling, the standard MC acceptance rule must be followed,

$$P_{\text{acc}} = \min\left[1, \frac{\rho_{\text{TPS}}\left(X^{(n)}\right) P_{\text{gen}}\left(X^{(n)} \to X^{(o)}\right)}{\rho_{\text{TPS}}\left(X^{(o)}\right) P_{\text{gen}}\left(X^{(o)} \to X^{(n)}\right)}\right] \quad (2.18)$$

Inserting the TPS path weight from Equation 2.13 via Equation 2.16 and assuming that the old path is valid, $h_{\text{TPS}}(X^{(o)}) = 1$,

$$P_{\text{acc}} = h_{\text{TPS}}\left(X^{(n)}\right) \min\left[1, \frac{\rho\left(x^{(n)}\right) P_{\text{PD}}(X^{(n)}|x^{(n)}) P_{\text{gen}}\left(X^{(n)} \to X^{(o)}\right)}{\rho\left(x^{(o)}\right) P_{\text{PD}}(X^{(o)}|x^{(o)}) P_{\text{gen}}\left(X^{(o)} \to X^{(n)}\right)}\right] \quad (2.19)$$

Inserting the shooting generation probabilities from Equation 2.17 and canceling the P_{PD} terms,

$$P_{\text{acc}} = h_{\text{TPS}}\left(X^{(n)}\right) \min\left[1, \frac{\rho\left(x^{(n)}\right) P_{\text{sel}}\left(x^{(n)} \mid X^{(n)}\right) P_{\text{vel}}\left(x^{(n)} \to x^{(o)}\right)}{\rho\left(x^{(o)}\right) P_{\text{sel}}\left(x^{(o)} \mid X^{(o)}\right) P_{\text{vel}}\left(x^{(o)} \to x^{(n)}\right)}\right] \quad (2.20)$$

The common way to generate new velocities is through drawing from the Maxwellian distribution,

$$P_{\text{vel}}\left(x^{(o)} \to x^{(n)}\right) = \rho\left(v^{(n)}\right) \quad (2.21)$$

and letting the phase point weight equal the product of the configuration and velocity weights $\rho(x)=\rho(r)\rho(v)$,

$$P_{\text{acc}} = h_{\text{TPS}}\left(X^{(n)}\right)\min\left[1,\frac{P_{\text{sel}}\left(x^{(n)}\mid X^{(n)}\right)\rho\left(r^{(n)}\right)\rho\left(v^{(n)}\right)\rho\left(v^{(o)}\right)}{P_{\text{sel}}\left(x^{(o)}\mid X^{(o)}\right)\rho\left(r^{(o)}\right)\rho\left(v^{(o)}\right)\rho\left(v^{(n)}\right)}\right] \tag{2.22}$$

Cancelling all the weights ($r^{(o)}=r^{(n)}$) and also recognizing that $P_{\text{sel}}\left(x|X\right)$ is just the probability of selecting one out of X's phase points except for the first and last (hence - 2),

$$P_{\text{acc}}\left(X^{(o)}\rightarrow X^{(n)}\right)=h_{\text{TPS}}\left(X^{(n)}\right)\min\left[1,\frac{\operatorname{len}\left(X^{(o)}\right)-2}{\operatorname{len}\left(X^{(n)}\right)-2}\right] \tag{2.23}$$

According to this rule, $X^{(n)}$ will always accepted if valid and has a path length that is equal or is less than $X^{(o)}$. Otherwise, the path will be accepted or rejected based on a probability. Through cycles of shooting moves, a statistical ensemble of reactive paths is obtained that obey the true dynamics arising from the studied system.

2.3.2 Replica Exchange Transition Interface Sampling

While TPS manages to efficiently sample reactive paths that provide dynamical insight, standard TPS (although versions exist) does not sample trajectories that are interesting and transitionary, but also unreactive. To enable such possibilities, a capable Dutch PhD student in the PD group cooks up the Replica Exchange Transition Interface Sampling (RETIS) method. In comparison to TPS, (RE)TIS defines a set of order parameter values $\{\lambda_0\ldots\lambda_i\ldots\lambda_N\}$ called interfaces, and is distributed between the space spanned by state A, $\lambda_A=\lambda_0$ and state B, $\lambda_B=\lambda_N$ with $\lambda_i<\lambda_{i+1}$. Each interface λ_i, except λ_N, defines a path ensemble denoted $[i^+]$ with an accompanying path weight,

$$\rho_{[i^+]}(X)=h_{[i^+]}\left(X\right)\rho\left(x_0\right)P_{PD}(X|x_0) \tag{2.24}$$

with $h_{[i^+]}\left(X\right)$,

$$h_{[i^+]}\left(X\right)=\begin{cases}1 & \text{if }\max(X)>\lambda_i\text{ and }x_0\in\text{A and }x_M\in(\text{A}\cup\text{B})\\ & \text{and }\{x_1\ldots x_{M-1}\}\notin(\text{A}\cup\text{B})\\ 0 & \text{otherwise}\end{cases} \tag{2.25}$$

with $\max(X)$ being the maximum order parameter reached by the path X. In comparison to the valid paths sampled in the TPS ensemble, TIS allows for the sampling of unreactive paths, paths that start and end in λ_0, as long as λ_i is crossed (see Figure 2.3.2 **Right**). The conditional crossing probability of the sampled paths in ensemble $[i^+]$ that also cross the next interface λ_{i+1}, $P\left(\lambda_{i+1}|\lambda_i\right)$, can be estimated from running cycles of shooting moves in the TIS path ensemble $[i^+]$,

$$P\left(\lambda_{i+1}|\lambda_i\right)=\frac{1}{L}\sum_{j=1}^{L}h_{[(i+1)^+]}\left(X_j\right) \tag{2.26}$$

L is the number of sampled paths. In order to estimate the flux through interface λ_0, an additional path simulation denoted $[0^-]$ (zero minus) must be performed. In contrast to the *plus* ensembles $[i^+]$, ensemble $[0^-]$ samples the inner state A itself, with the following TIS indicator requirement,

$$h_{[0^-]}(X) = \begin{cases} 1 & \text{if } \{x_0, x_M\} \notin \text{A and } \{x_1 \dots x_{M-1}\} \in \text{A} \\ 0 & \text{otherwise} \end{cases} \tag{2.27}$$

Together with $[0^+]$, the flux can be estimated from the average time (path length) spent on each side of λ_0,

$$f_A = \frac{1}{(\langle \text{len}([0^+]) \rangle + \langle \text{len}([0^-]) \rangle - 4)\,\Delta t} \tag{2.28}$$

with -4 accounting for overcounting the first and last frames for paths in both ensembles. Finally, the rate constant can be estimated to equal

$$k_{AB} = f_A P_A(\lambda_B \mid \lambda_A) = f_A \prod_{i=0}^{N-1} P_A(\lambda_{i+1} \mid \lambda_i) \tag{2.29}$$

The system and energy barrier (steepness) play a part in determining the number of ensembles/λ_i interfaces required to discretize the space between states A and B, where a higher number means more individual independent TIS simulations must be run to eventually obtain converged results for the rate constant calculation. A way to speed up the simulation is to employ replica exchange between two ensembles $[i^+]$ and $[(i+1)^+]$,

$$P_{\text{acc}}\left(X_a \in [i^+] \leftrightarrow X_b \in [(i+1)^+]\right) = \min\left[1, \frac{\rho_{[i^+]}(X_b)\,\rho_{[(i+1)^+]}(X_a)}{\rho_{[i^+]}(X_a)\,\rho_{[(i+1)^+]}(X_b)}\right] \tag{2.30}$$

By substituting in the path weight expressions, all terms except $h_{[(i+1)^+]}(X_a)$ cancel out, leaving,

$$P_{\text{acc}}\left(X_a \in [i^+] \leftrightarrow X_b \in [(i+1)^+]\right) = h_{[(i+1)^+]}(X_a) \tag{2.31}$$

which means that the RE move would be accepted if path X_a crosses λ_{i+1} (see Figure 2.3.3 **Left**. Compared to the shooting move that require time to run the PD simulations backwards and forward and forward in time, Equation 2.30 is the only calculation required to complete a RE move as the paths to be swapped have already been generated.

An alternative swapping scheme is required for the swap between ensembles $[0^-]$ and $[0^+]$ as they sample paths in opposite directions of interface λ_0. One way is through the so-called *point exchange* move, illustrated in Figure 2.3.3 **Right**. The move works by propagating the last and first point of $[0^-]$ and $[0^+]$ paths forward and backward until the interface λ_i is hit again. In this formulation, all point exchange moves should be accepted according to the standard acceptance rule. Compared to the standard RE move, the point exchange move require the running of PD to complete.

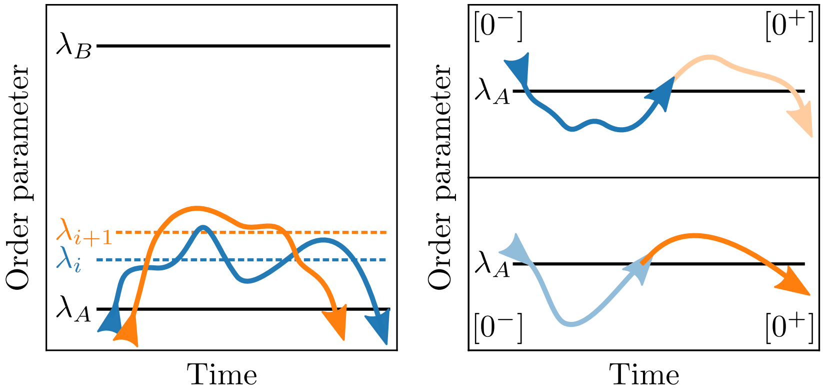


Figure 2.3.3: Left: The last accepted paths for ensemble $[i^+]$ and $[(i+1)^+]$, with both paths being valid in both ensembles as both cross λ_{i+1}. **Right:** The point exchange move for the ensembles $[0^+]$ and $[0^-]$. Here the light blue and light orange paths are integrated forward and backward to produce new orange and blue for $[0^+]$ and $[0^-]$.

Of course, TPS, TIS and RETIS are readily applied to MD simulations in order to study and calculate the rates of rare events occurring at the molecular scale. While this *panda dynamics* example is fictive, and path simulation techniques were originally derived for the rare event problem in molecular dynamics, a type of path sampling algorithm has in fact been applied to sample the dynamical game space spanned by chess [56]. Therefore, path sampling simulations may have its uses in other fields as well.

CHAPTER

THREE

ACCELERATING PATH SAMPLING CONVERGENCE

This chapter summarizes my main contributions to increasing the RETIS convergence speed. While the previous chapter explained theory through fictive analogies, the rest of this thesis, including this chapter and the included papers deals with the rare event problem occurring in molecular dynamics simulations.

3.1 Subtrajectory Monte Carlo Moves

The new path produced by a successful shooting move is correlated with the old path by sharing at least one common shooting/configuration point, as seen in Figure 2.3.2. To increase convergence and efficiency, fast decorrelating shooting moves called Stone Skipping (SS) and Web Throwing (WT) were introduced in [57]. Through generating a series of subtrajectories, these moves produce a new path that has no common configuration point with the old path. Alleviating the SS and WT *one-step crossing* issue, a third subtrajectory based move called Wire Fencing (WF) is introduced and detailed in paper B. The SS and WF moves are shown in Figure 3.1.1,

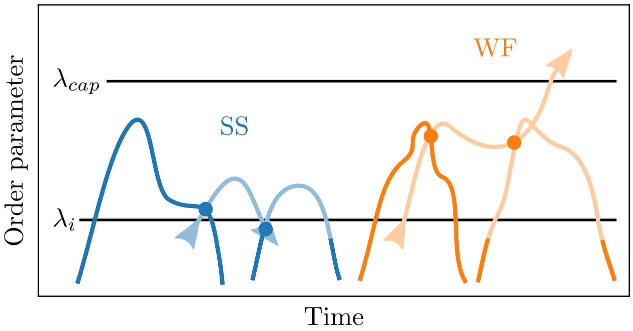


Figure 3.1.1: The SS move (in blue) and WF move (in orange). The moves in both cases have been run using a user-defined subtrajectory number of $N_s = 2$, where the last accepted subtrajectory has been extended to produce path *new*. In WF, an additional interface cap $\lambda_{\text{cap}} \leq \lambda_B$ can be defined to restrict the allowable subtrajectory propagation and shooting selection region.

One simple way to image the correlation reduction is to estimate each subtrajectory to have been extended and produced by standard shooting. For example, a WF move with $N_s = 10$ subtrajectory generations can be viewed as running 10 shooting moves, but with the benefit of not having to extend the 9 shooting moves below λ_i and above λ_{cap}. To preserve correct statistical sampling, the SS/WT/WF acceptance rules have been derived utilizing the super-detailed balance equation,

$$\begin{aligned}\rho_{[i^+]}\left(X^{(o)}\right) P_{\text{gen}}\left(X^{(o)} \to X^{(n)} \text{ via } \chi\right) P_{\text{acc}}\left(X^{(o)} \to X^{(n)} \text{ via } \chi\right) = \\ \rho_{[i^+]}\left(X^{(n)}\right) P_{\text{gen}}\left(X^{(n)} \to X^{(o)} \text{ via } \bar{\chi}\right) P_{\text{acc}}\left(X^{(n)} \to X^{(o)} \text{ via } \bar{\chi}\right)\end{aligned} \tag{3.1}$$

With the following acceptance rule

$$P_{\text{acc}} = h_{[i^+]}\left(X^{(n)}\right) \min\left[1, \frac{\rho_{\lambda_i}\left(X^{(n)}\right) P_{\text{gen}}\left(X^{(n)} \to X^{(o)} \text{ via } \bar{\chi}\right)}{\rho_{\lambda_i}\left(X^{(o)}\right) P_{\text{gen}}\left(X^{(o)} \to X^{(n)} \text{ via } \chi\right)}\right] \tag{3.2}$$

where χ (and $\bar{\chi}$) denotes the specific (reverse) construction pathway from $X^{(o)}$ to $X^{(n)}$. Specific SS/WT/WF acceptance rules can be derived by inserting their P_{gen} equation. A simplified WF P_{gen} can be explained in words,

1. From path *old*, select a valid subpath s^0, a subpath starting and ending in interfaces $\lambda_i \to \lambda_i$, $\lambda_i \to \lambda_{\text{cap}}$ or $\lambda_{\text{cap}} \to \lambda_i$, but not $\lambda_{\text{cap}} \to \lambda_{\text{cap}}$. $\lambda_i < \lambda_{\text{cap}} \leq \lambda_B$ is a user-defined variable.

2. Perform N_s shooting moves starting with s^0 in an alternative ensemble with states A and B having interfaces λ_i and λ_{cap} and where only λ_{cap}-λ_{cap} subpaths are rejected. This does not imply a rejection of the overall MC move.

3. Extend the last accepted subtrajectory to end in interfaces λ_A and/or λ_B. Select a time direction with a 50% probability to obtain path *new*.

4. In the case of path *new* starting and ending in λ_B-λ_B or λ_B-λ_A, the path is automatically rejected. Otherwise, path *new* is accepted or rejected based on the acceptance rule.

An expression common to SS/WT/WF is obtained when inserting their P_{gen} into the super-detailed balance acceptance rule [35],

$$\frac{P_{\text{gen}}\left(X^{(n)} \to X^{(o)} \text{ via } \bar{\chi}\right)}{P_{\text{gen}}\left(X^{(o)} \to X^{(n)} \text{ via } \chi\right)} = \frac{\rho_{[i^+]}\left(X^{(o)}\right)/M_{[i^+]}(X^{(n)})}{\rho_{[i^+]}\left(X^{(n)}\right)/M_{[i^+]}(X^{(o)})} \tag{3.3}$$

with $M_{[i^+]}(X)$ being a number dependent on the path, ensemble, and subtrajectory move (SS/WT/WF). Common to all the moves, the $M_{[i^+]}(X)$ value is likely to increase with path length. Inserting Equation 3.3 into the acceptance rule and assuming $h_{[i^+]}\left(X^{(n)}\right) = 1$,

$$P_{\text{acc}} = \min\left[1, \frac{\rho_{[i^+]}\left(X^{(n)}\right)\rho_{[i^+]}\left(X^{(o)}\right) M_{[i^+]}(X^{(o)})}{\rho_{[i^+]}\left(X^{(o)}\right)\rho_{[i^+]}\left(X^{(n)}\right) M_{[i^+]}(X^{(n)})}\right] = \min\left[1, \frac{M_{[i^+]}(X^{(o)})}{M_{[i^+]}(X^{(n)})}\right] \tag{3.4}$$

Detailed derivations for SS and WT are shown in the SI of [57], and for WF in paper B [35].

3.1.1 High Acceptance

Compared to regular shooting, the total number of MD steps required to complete a SS/WT/WF path can possibly be much higher due to the generation of subtrajectories. While the compensation is higher decorrelation, a path rejection also means wasting more MD steps. To maximize acceptance, rejected $\lambda_B \rightarrow \lambda_A$ paths can instead be accepted by reversing the path velocities to form valid $\lambda_A \rightarrow \lambda_B$ ensemble paths. The 50% probability in selecting time direction thus disappears for reactive paths and P_{gen} must as a consequence be multiplied by a factor $q(X)$,

$$q(X) = \begin{cases} 1 & \text{if } X \in \{A \rightarrow A\} \\ 2 & \text{if } X \in \{A \rightarrow B\} \end{cases} \tag{3.5}$$

Additionally, paths can be sampled according to alternative ensemble definitions that bring the acceptance probability in Equation 3.4 to equal unity. Again assuming $h_{[i^+]}\left(X^{(n)}\right) = 1$,

$$P_{\text{acc}} = \min\left[1, \frac{\rho_{\tilde{[i^+]}}\left(X^{(n)}\right) q\left(X^{(o)}\right) P_{\text{gen}}\left(X^{(n)} \rightarrow X^{(o)} \text{ via } \bar{\chi}\right)}{\rho_{\tilde{[i^+]}}\left(X^{(o)}\right) q\left(X^{(n)}\right) P_{\text{gen}}\left(X^{(o)} \rightarrow X^{(n)} \text{ via } \chi\right)}\right] \tag{3.6}$$

with $\tilde{\rho}_{[i^+]}$,

$$\tilde{\rho}_{[i^+]}(\text{ X }) = \rho_{[i^+]}(\text{ X })w_{[i^+]}(\text{ X }) \tag{3.7}$$

and

$$w_{[i^+]}(\text{X}) = q(\text{X})M_{[i^+]}(\text{X}) \tag{3.8}$$

with Equation 3.3, P_{acc} becomes:

$$\begin{aligned} P_{\text{acc}} =& \min\left[1, \frac{\rho_{\tilde{[i^+]}}\left(X^{(n)}\right) q\left(X^{(o)}\right) \rho_{[i^+]}\left(X^{(o)}\right) M_{\lambda_i}(X^{(o)})}{\rho_{\tilde{[i^+]}}\left(X^{(o)}\right) q\left(X^{(n)}\right) \rho_{[i^+]}\left(X^{(n)}\right) M_{[i^+]}(X^{(n)})}\right] \\ =& \min\left[1, \frac{\rho_{[i^+]}\left(X^{(n)}\right) M_{[i^+]}(X^{(n)})q(X^{(n)})q\left(X^{(o)}\right) \rho_{[i^+]}\left(X^{(o)}\right) M_{[i^+]}(X^{(o)})}{\rho_{[i^+]}\left(X^{(o)}\right) M_{[i^+]}(X^{(o)})q\left(X^{(o)}\right) q\left(X^{(n)}\right) \rho_{[i^+]}\left(X^{(n)}\right) M_{[i^+]}(X^{(n)})}\right] \\ =& 1 \end{aligned} \tag{3.9}$$

The only possibility of rejection will therefore be when a λ_B-λ_B path is generated, as $h_{[i^+]}\left(X^{(n)}\right) = 0$. This combination of methods to minimize rejection can be called high acceptance (HA). To obtain correct statistics, the sampled paths can be reweighed in the post-simulation analysis,

$$P\left(\lambda_{i+1}|\lambda_i\right) = \frac{\sum_j w_{[i^+]}^{-1}\left(X_j\right) h_{[(i+1)^+]}\left(X_j\right)}{\sum_j w_{[i^+]}^{-1}\left(X_j\right)} \tag{3.10}$$

A penalty for HA, however, is the reduction from 100% replica exchange acceptance,

$$P_{\text{acc}} = h_{[(i+1)^+]}\left(X_a\right) \times \min\left[1, \frac{w_{[i^+]}(X_b)w_{[(i+1)^+]}(X_a)}{w_{[i^+]}(X_a)w_{[(i+1)^+]}(X_b)}\right] \tag{3.11}$$

This is especially impactful if high acceptance is enabled for the zero ensembles $[0^-]$ and $[0^+]$ as the point exchange move requires MD. However, since a subpath in the zero ensembles has the same length as standard paths, running subtrajectory moves does not provide much benefit in the first place. Rather, a general setup is to run standard shooting moves in the zero ensembles and SS/WT/WF in the other plus ensembles. Note that while the swapping acceptance probability is reduced for HA ensembles, the reduction is partly migrated by HA also indiscriminately accepting long paths. Since long paths contain more phase points, $h_{[(i+1)^+]}$ can be more likely to equal 1 with HA than without.

3.2 Infinite Replica Exchange

As mentioned previously, the general practice is to set the probability to perform either a cycle of RE or a cycle of compute moves (CP) to around $P_{\text{RE}} = 0.5$ for standard REMC simulations. A CP move here refers to one of the shooting/SS/WT/WF moves. This not-too-small, not-too-large P_{RE} value allows frequent replica exchange attempts to occur without too many that can possibly slow down the overall sampling simulation. Recent developments [16, 17, 58, 59, 60], however, show that excluding overhead time, the convergence rate theoretically increases with increasing swapping probability, with the fastest convergence rate occurring when the swapping probability approaches one ($P_{\text{RE}} \rightarrow 1$). This convergence increase can be illustrated through the occurrence of two successive cycles of CP moves in a REMC simulation with a finite swapping probability $P_{\text{RE}} = 0.5$. For example, given a RETIS path simulation with two high-acceptance ensembles $[1^+]$ and $[2^+]$ having initial paths a_0 and b_0, then two cycles of successive CP moves and two failed RE attempts would result in a sequence of paths $\{\mathrm{a}_0, \mathrm{a}_1, \mathrm{a}_2, \mathrm{a}_2, \mathrm{a}_2\}$ being sampled in $[1^+]$ and $\{\mathrm{b}_0, \mathrm{b}_1, \mathrm{b}_2, \mathrm{b}_2, \mathrm{b}_2\}$ being sampled in $[2^+]$, as also shown in Table 3.2.1,

Table 3.2.1: A REMC scheme illustrating the development of states a_0 and b_0 in ensembles $[1^+]$ and $[2^+]$ with cycles of CP and RE moves. States a_0 and b_0 among others are *pushed out* by CP moves, disabling the possibility of being sampled in each other's ensemble.

		CP		CP		RE		RE		CP	
$[1^+]$	a_0	$\rightarrow$	a_1	$\rightarrow$	a_2	$\rightarrow$	a_2	$\rightarrow$	a_2	$\rightarrow$	a_3
$[2^+]$	b_0	$\rightarrow$	b_1	$\rightarrow$	b_2	$\rightarrow$	b_2	$\rightarrow$	b_2	$\rightarrow$	b_3

Even with possibly high statistical weights in each other's ensembles, the absence of RE moves between the initial CP moves prevents the states $\{\mathrm{b}_0, \mathrm{b}_1\}$ from also being sampled in ensemble $[1^+]$ and vice versa, as the paths get immediately *pushed out* by the CP moves generating paths b_2 and b_3. Additionally, in the case where replica exchange moves do occur between two CP moves, then one or a series of swapping attempts might not be enough to produce an accepted replica exchange if a_2 and b_2 has low but nonzero statistical weights in the other ensemble. Since RE moves are essentially cost-free compared to the expensive CP moves, a high P_{RE} value would allow access to additional, possibly unsampled paths for free. Therefore, if the replica exchange probability is taken to the upper limit, $P_{\text{RE}} \rightarrow 1$, then all active states in all ensembles will be perfectly sampled through an infinite number of replica exchange attempts between two cycles of CP moves. For example, if two paths a_0 and b_0 are valid in ensemble $[1^+]$ but only b_0 is valid in ensemble $[2^+]$, then the RE swap will always fail as $h_{[2^+]}(\mathrm{a}_0) = 0$ resulting in the following sampling pattern in ensembles $[1^+]$ and $[2^+]$,

$$\begin{aligned}
&[1^+] : \{\mathrm{a}_0, \mathrm{a}_0, \mathrm{a}_0, \mathrm{a}_0, \mathrm{a}_0, \mathrm{a}_0, \dots\} \\
&[2^+] : \{\mathrm{b}_0, \mathrm{b}_0, \mathrm{b}_0, \mathrm{b}_0, \mathrm{b}_0, \mathrm{b}_0, \dots\}
\end{aligned} \tag{3.12}$$

After *one* infinite swap, the paths sampled by $[1^+]$ will 100% be a_0 and b_0 for $[2^+]$. Given that the HA weights w for a_0 and b_0 both equal 1, then the non-converged average of some property $\langle A \rangle$ can be initially estimated as

$$\begin{aligned}
\langle A \rangle_{[1^+]} =& \frac{M \cdot A(\mathrm{a}_0)}{M} \\
\langle A \rangle_{[2^+]} =& \frac{M \cdot A(\mathrm{b}_0)}{M}
\end{aligned} \tag{3.13}$$

where $M \to \infty$ is the number of sampled states from *one* infinity swap attempts. Assuming we now perform one cycle of CP moves that result in acceptances $\mathrm{a}_0 \to \mathrm{a}_1$ and $\mathrm{b}_0 \to \mathrm{b}_1$. Let paths a_1 and b_1 be valid in both ensembles and also have HA weights equal 1, resulting in $P_{\text{acc}} = 1$ for both swapping directions $\mathrm{a}_{1\in[1^+]} \leftrightarrow \mathrm{b}_{1\in[2^+]}$ and $\mathrm{b}_{1\in[1^+]} \leftrightarrow \mathrm{a}_{1\in[2^+]}$ so that the sampled paths follow the consequent sampling pattern,

$$\begin{aligned}
&[1^+] : \{\dots, \mathrm{a}_1, \mathrm{b}_1, \mathrm{a}_1, \mathrm{b}_1, \mathrm{a}_1, \mathrm{b}_1, \dots\} \\
&[2^+] : \{\dots, \mathrm{b}_1, \mathrm{a}_1, \mathrm{b}_1, \mathrm{a}_1, \mathrm{b}_1, \mathrm{a}_1, \dots\}
\end{aligned} \tag{3.14}$$

resulting in states a_1 and b_1 having a fraction of $M/2$ in both ensembles $[1^+]$ and $[2^+]$ after *one* infinite swap,

$$\begin{aligned}
\langle A \rangle_{[1^+]} =& \frac{M \cdot A(\mathrm{a}_0) + \frac{M}{2} \cdot A(\mathrm{a}_1) + \frac{M}{2} \cdot A(\mathrm{b}_1)}{M + M} \\
\langle A \rangle_{[2^+]} =& \frac{M \cdot A(\mathrm{b}_0) + \frac{M}{2} \cdot A(\mathrm{a}_1) + \frac{M}{2} \cdot A(\mathrm{b}_1)}{M + M}
\end{aligned} \tag{3.15}$$

Repeating the ∞RE and CP loop, the next step is to perform a set of CP moves again. However, which of the two paths a_1 and b_1 is the last accepted path to be used for the CP move in $[1^+]$? The last sampled path now becomes probabilistic and depends on the fraction of all the sampled paths in one infinity swap. In this case, there is a equally likely chance to shoot from a_1 and b_1 for both ensembles. Given that a_1 is selected for ensemble $[1^+]$, then two successful CP moves gives for $[1^+]$, $\mathrm{a}_1 \to \mathrm{a}_2$ and for $[2^+]$, $\mathrm{b}_1 \to \mathrm{b}_2$ where we again assume that paths a_2 and b_2 are both valid in both ensembles. In this case, the paths have a certain ensemble $w_{[i^+]}(X)$ weight, which affects the swapping probability in Equation 3.11. Let us assume path a_2 has weights $w_{[1^+]}(\mathrm{a}_2) = 3$ and $w_{[2^+]}(\mathrm{a}_2) = 2$, and for b_2, $w_{[1^+]}(\mathrm{b}_2) = 4$ and $w_{[2^+]}(\mathrm{b}_2) = 1$. These weights can be arranged into a weight matrix W form,

$$W = \begin{array}{c} \\ \mathrm{a}_2 \\ \mathrm{b}_2 \end{array} \begin{array}{c} \begin{array}{cc} [1^+] & [2^+] \end{array} \\ \begin{bmatrix} 3 & 2 \\ 4 & 1 \end{bmatrix} \end{array} \tag{3.16}$$

letting the swapping yield the following acceptance probabilities,

$$P_{\text{acc}}\left(\mathrm{a}_{2\in[1^+]} \leftrightarrow \mathrm{b}_{2\in[2^+]}\right) = \min\left[1, \frac{w_{[1^+]}\left(\mathrm{b}_2\right) w_{[2^+]}\left(\mathrm{a}_2\right)}{w_{[1^+]}\left(\mathrm{a}_2\right) w_{[2^+]}\left(\mathrm{b}_2\right)}\right] = \min\left[1, \frac{4 \cdot 2}{3 \cdot 1}\right] = 1 \tag{3.17}$$

and the reverse,

$$P_{\text{acc}}\left(\mathrm{b}_{1\in[1^+]} \leftrightarrow \mathrm{a}_{1\in[2^+]}\right) = \min\left[1, \frac{w_{[1^+]}\left(\mathrm{a}_2\right) w_{[2^+]}\left(\mathrm{b}_2\right)}{w_{[1^+]}\left(\mathrm{b}_2\right) w_{[2^+]}\left(\mathrm{a}_2\right)}\right] = \min\left[1, \frac{3 \cdot 1}{4 \cdot 2}\right] = \frac{3}{8} \tag{3.18}$$

resulting into biased randomised sampling,

$$\begin{aligned} [1^+] &: \{\dots, \mathrm{a}_2, \mathrm{b}_2, \mathrm{b}_2, \mathrm{b}_2, \mathrm{a}_2, \mathrm{b}_2, \dots\} \\ [2^+] &: \{\dots, \mathrm{b}_2, \mathrm{a}_2, \mathrm{a}_2, \mathrm{a}_2, \mathrm{b}_2, \mathrm{a}_2, \dots\} \end{aligned} \tag{3.19}$$

In this case, what would the fraction of sampled paths a_2 and b_2 be in ensembles $[1^+]$ and $[2^+]$ after *one* infinite swap? This perhaps non-trivial solution can be obtained through the combinatorics of possible path-ensemble permutations, with permutations being denoted as σ forming a set C. In the case of two paths and two ensembles, only two permutations are possible, $\sigma_1 : \mathrm{a}_{2\in[1^+]}, \mathrm{b}_{2\in[2^+]}$ and $\sigma_2 : \mathrm{b}_{2\in[1^+]}, \mathrm{a}_{2\in[2^+]}$. The statistical weight of a permutation is represented by the product weights of its members,

$$\begin{aligned} \rho\left(\sigma_1\right) =& \rho\left(\mathrm{a}_{2\in[1^+]}, \mathrm{b}_{2[2^+]}\right) = \rho_{[1^+]}\left(\mathrm{a}_2\right)\rho_{[2^+]}\left(\mathrm{b}_2\right) = 3 \cdot 1 = 3 \\ \rho\left(\sigma_2\right) =& \rho\left(\mathrm{b}_{2\in[1^+]}, \mathrm{a}_{2[2^+]}\right) = \rho_{[1^+]}\left(\mathrm{b}_2\right)\rho_{[2^+]}\left(\mathrm{a}_2\right) = 4 \cdot 2 = 8 \end{aligned} \tag{3.20}$$

The sampled fraction/probability of a_2 in $[1^+]$ after an ∞RE can therefore be expressed as σ, the weights of the possible permutations, with a_2 in $[1^+]$, $\sigma \in C_{\mathrm{a}_2\in[1^+]}$ divided by all the permutations possible, $\sigma \in C$,

$$p_{11} = \frac{\sum_{\sigma\in C_{\mathrm{a}_2\in[1^+]}} \rho(\sigma)}{\sum_{\sigma\in C} \rho(\sigma)} = \frac{3}{3+8} = \frac{3}{11} \tag{3.21}$$

resulting in the following probability matrix P,

$$P = \begin{array}{c} \\ \mathrm{a}_1 \\ \mathrm{b}_1 \end{array} \begin{array}{c} \begin{array}{cc} [1^+] & [2^+] \end{array} \\ \begin{bmatrix} 3 & 8 \\ 8 & 3 \end{bmatrix} \end{array} \frac{1}{11}$$

Now 5 unique paths are sampled in each ensemble after *three* infinite swaps separated by shooting moves,

$$\begin{aligned} \langle A \rangle_{[1^+]} =& \frac{M \cdot A\left(\mathrm{a}_0\right) + \frac{M}{2} \cdot A\left(\mathrm{a}_1\right) + \frac{M}{2} \cdot A\left(\mathrm{b}_1\right) + \frac{3M}{11\cdot 3} \cdot A\left(\mathrm{a}_2\right) + \frac{8M}{11\cdot 2} \cdot A\left(\mathrm{b}_2\right)}{M + M + \frac{3M}{11\cdot 3} + \frac{8M}{11\cdot 2}} \\ \langle A \rangle_{[2^+]} =& \frac{M \cdot A\left(\mathrm{b}_0\right) + \frac{M}{2} \cdot A\left(\mathrm{a}_1\right) + \frac{M}{2} \cdot A\left(\mathrm{b}_1\right) + \frac{8M}{11\cdot 4} \cdot A\left(\mathrm{a}_2\right) + \frac{3M}{11\cdot 1} \cdot A\left(\mathrm{b}_2\right)}{M + M + \frac{8M}{11\cdot 4} + \frac{3M}{11\cdot 1}} \end{aligned} \tag{3.22}$$

where the $1/w$ weights are applied to a_2 and b_2 to counteract the distorted HA ensemble distribution. For the previous samples, $1/w = 1$ in this hypothetical simulation. As M gets canceled on both sides, the average becomes expressed through the probability coefficients p_{ij} of sampled paths produced *per* infinite swap,

$$\begin{aligned}\langle A\rangle_{[1^+]} &= \frac{A\left(\mathrm{a}_0\right)+\frac{1}{2}\cdot A\left(\mathrm{a}_1\right)+\frac{1}{2}\cdot A\left(\mathrm{b}_1\right)+\frac{3}{11\cdot 3}\cdot A\left(\mathrm{a}_2\right),\frac{8}{11\cdot 2}\cdot A\left(\mathrm{b}_2\right)+\cdots}{1+1+\frac{3}{11\cdot 3}+\frac{8}{11\cdot 2}\cdots}\\ \langle A\rangle_{[2^+]} &= \frac{A\left(\mathrm{b}_0\right)+\frac{1}{2}\cdot A\left(\mathrm{a}_1\right)+\frac{1}{2}\cdot A\left(\mathrm{b}_1\right)+\frac{8}{11\cdot 4}\cdot A\left(\mathrm{a}_2\right),\frac{3}{11\cdot 1}\cdot A\left(\mathrm{b}_2\right)+\cdots}{1+1+\frac{8}{11\cdot 4}+\frac{3}{11\cdot 1}+\cdots}\end{aligned} \tag{3.23}$$

Another possibility to occur if this example is to be continued, is that CP moves can of course be rejected, $\mathrm{a}_2 \rightarrow \mathrm{a}_2$, causing a_2 to participate in multiple infinite swaps and thus accumulate probabilities in the average calculation until the state is eventually *pushed out* by a successful CP move $\mathrm{a}_2 \rightarrow \mathrm{a}_3$. If the ∞REMC example is instead extended to have 4 ensembles, then a resultant 4 dimensional P matrix can be expressed in Table 3.2.2,

Table 3.2.2: An ∞REMC swapping scheme with four ensembles showing a cycle of CP moves followed by infinite replica exchange for four ensembles and paths. The samples are initially multiplied with zero because their "weight" in average calculations reduces to zero as an additional "infinite" amount of samples are added from an ∞RE move. Note here that all the p_{0j} and p_{i0} coefficients except p_{00} are zero as b_0-d_0 cannot be swapped into $[0^-]$ vice versa for a_0.

	CP	∞RE		CP
$[0^-]$	$\rightarrow\ 0\cdot\mathrm{a}_0$	$\rightarrow$	$p_{00}\cdot\mathrm{a}_0+p_{01}\cdot\mathrm{b}_0+p_{02}\cdot\mathrm{c}_0+p_{03}\cdot\mathrm{d}_0$	$\rightarrow$
$[0^+]$	$\rightarrow\ 0\cdot\mathrm{b}_0$	$\rightarrow$	$p_{10}\cdot\mathrm{a}_0+p_{11}\cdot\mathrm{b}_0+p_{12}\cdot\mathrm{c}_0+p_{13}\cdot\mathrm{d}_0$	$\rightarrow$
$[1^+]$	$\rightarrow\ 0\cdot\mathrm{c}_0$	$\rightarrow$	$p_{20}\cdot\mathrm{a}_0+p_{21}\cdot\mathrm{b}_0+p_{22}\cdot\mathrm{c}_0+p_{23}\cdot\mathrm{d}_0$	$\rightarrow$
$[2^+]$	$\rightarrow\ 0\cdot\mathrm{d}_0$	$\rightarrow$	$p_{30}\cdot\mathrm{a}_0+p_{31}\cdot\mathrm{b}_0+p_{32}\cdot\mathrm{c}_0+p_{33}\cdot\mathrm{d}_0$	$\rightarrow$

The calculation of P from W enables an ∞REMC scheme without the overhead of explicitly carrying out infinite swaps. A problem, however, is the factorial scaling of the P calculation with the number of ensembles in the REMC simulation [16]. This harsh scaling generally prevents the application of ∞REMC with too many ensembles as a high number can cause a P calculation to take up the majority of the simulation time. Paper A, however, realizes that the problem of calculating P can be reformulated into the calculation of permanents, where each P matrix element p_{ij} can be calculated using Equation 3.24,

$$p_{ij} = \frac{w_{ij}\,\mathrm{perm}(w_{\{ij\}})}{\mathrm{perm}(W)} \tag{3.24}$$

where $W_{\{ij\}}$ is the W matrix without row i and column j. A permanent (of a matrix) is the same as a determinant, except with only additions,

$$\mathrm{perm}\begin{bmatrix} A & B & C\\ D & E & F\\ G & H & I\end{bmatrix} = \begin{matrix} A\cdot(E\cdot I+H\cdot F)+\\ B\cdot(D\cdot I+F\cdot G)+\\ C\cdot(D\cdot H+E\cdot I)\end{matrix} \tag{3.25}$$

By the use of fast permanent algorithms [61, 62, 63], the scaling reduces from $O(n!)$ to $O\left(2^n\times n^2\right)$, which is enough to reduce a 20-dimensional P matrix calculation time from $\sim 15 \cdot 10^6$ years to ~ 711 seconds using a certain standard computer [36]. Often, however, ∞RETIS yield many zero elements in the W matrix due to paths not being valid in certain ensembles. Therefore, depending on the system studied, the permanent calculation of a big W matrix can be block diagonalized into a product of smaller permanents. In the case where no paths crosses the next interface, the W permanent can be expressed as the product of one-dimensional diagonal elements,

$$\text{perm}\begin{bmatrix} A & 0 & 0 \\ D & E & 0 \\ G & H & I \end{bmatrix} = \begin{matrix} A \cdot (E \cdot I + H \cdot 0) + \\ 0 \cdot (D \cdot I + F \cdot G) + \\ 0 \cdot (D \cdot H + E \cdot I) \end{matrix} = A \cdot E \cdot I \tag{3.26}$$

Depending on the studied system, ensemble numbers higher than 20 can be initiated. For example, in Paper C, the rare event of bubble nucleation in water is studied at boiling temperature with 80 ensembles. However, due to finite simulation size and the lack of nucleation sites, the resulting energy barrier becomes exceedingly steep and causes the paths sampled to rarely cross the next interface. The resulting W consequently consists of many zero elements, regularly allowing the W permanent calculation to be blocked down to a product of 1, 2 and 3-dimensional permanents. Note, however, that these simulations performed also utilized asynchronous RE, as explained in the next section, reducing the maximum possible W size to 41.

3.3 Asynchronous (Infinite) Replica Exchange

Introducing RE to TIS increases the convergence speed through higher sampling efficiency. In other words, the (CPU/GPU) computing efficiency increases. However, the introduction of RE can also slow down the wall time (elapsed real-time) efficiency. For example, say a certain RETIS simulation requires 4 ensembles to be employed. One cycle of CP moves will consequently demand 4 CP moves to be performed (and completed) in 4 ensembles before the next cycle of CP/RE moves can be started. Practically, the simplest and most standard way to do so would be to run the 4 CP moves sequentially, one after another using the hardware available, like one single computer node (Figure 3.3.1 **Top Right**). On the other hand for TIS simulations, each CP move can be run completely in parallel/independently using individual hardware for each ensemble, since no communication occurs between TIS ensembles (Figure 3.3.1 **Top Left**). Therefore, compared to standard (sequential) RETIS, TIS is fully parallelizable and can complete many more CP moves in the same amount of wall time. Of course, the 4 CP moves to be run by RETIS can also be run in parallel using 4 nodes, reducing the total wall time from the sum of 4 sequential CP moves to only the slowest completing CP move, as all moves must be completed in order for the next cycle of CP/RE to commence (Figure 3.3.1 **Bottom Left**). However, if the disparities in the CP completion time between the ensembles are large, as is often the case for path sampling simulations, then all but one of the nodes must remain idle for possibly hours until the last ensemble is finished. RETIS is consequently not very efficiently parallelizable, reducing the standard RETIS implementations (PyRETIS [64, 65], OPS [66]) to run fully sequentially. Therefore, while sequential RETIS maximizes compute efficiency, the "sequential lock" imposed by RE obfuscates the RETIS method from being the undisputed "overall best" of the two. Instead, running TIS over RETIS can be favorable in the case when hardware is of abundance, enabling better wall time efficiency with the cost of lower compute efficiency.

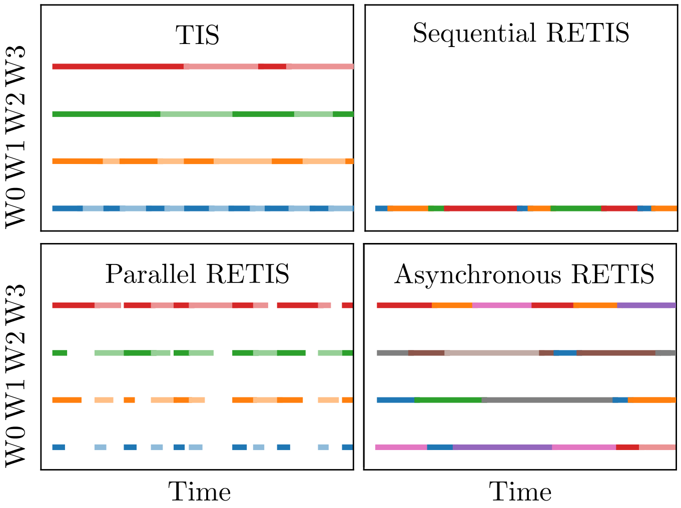


Figure 3.3.1: The hardware utilization of TIS and various RETIS flavors are plotted against time. Except for Asynchronous RETIS, all other simulations run four ensembles (color-coded). **Top Left:** Embarrassingly parallel TIS. **Top Right:** Parallel RETIS using only one worker, with the RE move occurring when the colored sequence repeats. **Bottom Left:** Sequential RETIS using one worker per ensemble. Due to the sequential lock, many of the workers have to remain idle before allowing to work again. **Bottom Right:** Asynchronous RETIS running 8 ensembles with 4 workers. At the time a worker finishes a CP move, RE occurs between the other free ensembles.

To keep the maximum compute efficiency from RE while allowing for high wall time efficiency through parallelization, the main authors of paper A derive a parallelizable RE scheme by utilizing an alternative, self-named "twisted" detailed balance relation. The resulting asynchronous and infinite RETIS scheme, ∞RETIS, allows for multiple standard MC moves to be run asynchronously in parallel while concurrently allowing for RE moves to occur between the other non-occupied ensembles (Figure 3.3.1 **Bottom Right**). To start, a number of *workers* (one processor unit, one node or a group of nodes) must be selected. The number of workers can be manually chosen depending on the hardware available, but has to be restricted to a number between 1 and the total number of defined ensembles. The algorithm starts with an initialization phase,

1. Schedule each worker to perform either a CP move for a path and an ensemble or a point exchange move if possible.

The ensembles and paths occupied by the workers are now unavailable for swapping. Then for each time a worker is done,

2. The new path (or pair of paths in case of the point exchange move) generated by the finished worker is accepted or rejected based on the standard acceptance rule. The ensemble(s) becomes unoccupied.

3. Calculate and record the probability P matrix from the resulting weight W matrix consisting of only the unoccupied ensembles.

4. The worker is assigned to work on a random, available ensemble j and a path i based on the probability p_{ij}. If j is either $[0^-]$ or $[0^+]$ and both ensembles are free, then there is a probability (usually 50%) to perform a point exchange move.

This loop continues until the results are converged. Since only free ensembles are available for swap moves, the matrix size equals the difference between the number of ensembles, N_{ens}, and the number of workers, N_{w}, plus either zero or one,

$$\text{size}(W) = N_{\text{ens}} - N_{\text{w}} + (0 \text{ or } 1) \tag{3.27}$$

depending on whether a busy worker occupies two ensembles or not. Since CP moves require variable time to complete, sampling is updated for all free ensembles whenever a worker is finished with their CP move.

To show how this algorithm works in practice, an ∞RETIS simulation with standard acceptance, two workers and 5 ensembles is now considered. To start, 5 valid paths have the following weights,

$$W = \begin{array}{c} \\ \mathrm{p}_0 \\ \mathrm{p}_1 \\ \mathrm{p}_2 \\ \mathrm{p}_3 \\ \mathrm{p}_4 \end{array} \begin{array}{c} \begin{array}{ccccc} [0^-] & [0^+] & [1^+] & [2^+] & [3^+] \end{array} \\ \left[\begin{array}{ccccc} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 \end{array}\right] \end{array} \tag{3.28}$$

As shown in W, p_0 is only a valid path in $[0^-]$ and the other paths p_{1-4} are only valid up till "its own" $[i^+]$ ensemble. All of the swapping attempts will be rejected, so P consequently results in the identity matrix,

$$P = \begin{array}{c} \\ \mathrm{p}_0 \\ \mathrm{p}_1 \\ \mathrm{p}_2 \\ \mathrm{p}_3 \\ \mathrm{p}_4 \end{array} \begin{array}{c} \begin{array}{ccccc} [0^-] & [0^+] & [1^+] & [2^+] & [3^+] \end{array} \\ \left[\begin{array}{ccccc} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{array}\right] \end{array} \tag{3.29}$$

reducing the possible paths to shoot from for a certain ensemble to one. The simulation can now be initiated by assigning worker w_0 to run a CP move with p_2 in $[1^+]$ and p_4 in $[3^+]$ for w_1. Now entering the main ∞RETIS loop and waiting until one of the workers has finished their CP move, the first worker to finish is w_0 with an accepted CP move $p_2 \rightarrow p_5$ in $[1^+]$. Say p_5 is valid up to ensemble $[2^+]$, then the W matrix can be expressed as,

$$W = \begin{array}{c} \\ \mathrm{p}_0 \\ \mathrm{p}_1 \\ \mathrm{p}_5 \\ \mathrm{p}_3 \\ \mathrm{p}_4 \end{array} \begin{array}{c} \begin{array}{ccccc} [0^-] & [0^+] & [1^+] & [2^+] & [3^+] \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & - \\ 0 & 1 & 0 & 0 & - \\ 0 & 1 & 1 & 1 & - \\ 0 & 1 & 1 & 1 & - \\ - & - & - & - & - \end{bmatrix} \end{array} \tag{3.30}$$

with the last row and column being blocked out by the busy worker w_1. The resulting P matrix can be calculated as,

$$P = \begin{array}{c} \\ \mathrm{p}_0 \\ \mathrm{p}_1 \\ \mathrm{p}_5 \\ \mathrm{p}_3 \\ \mathrm{p}_4 \end{array} \begin{array}{c} \begin{array}{ccccc} [0^-] & [0^+] & [1^+] & [2^+] & [3^+] \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & - \\ 0 & 1 & 0 & 0 & - \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & - \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & - \\ - & - & - & - & - \end{bmatrix} \end{array} \tag{3.31}$$

Since swapping is done, the free worker w_0 can be reassigned to perform a new task. In this case, a point exchange is to be run, requiring paths p_0 and p_1 in $[0^-]$ and $[0^+]$. After some time, w_1 is done with a failed CP move $p_4 \to p_4$ in $[3^+]$,

$$W = \begin{array}{c} \\ \mathrm{p}_0 \\ \mathrm{p}_1 \\ \mathrm{p}_5 \\ \mathrm{p}_3 \\ \mathrm{p}_4 \end{array} \begin{array}{c} \begin{array}{ccccc} [0^-] & [0^+] & [1^+] & [2^+] & [3^+] \end{array} \\ \begin{bmatrix} - & - & - & - & - \\ - & - & - & - & - \\ - & - & 1 & 1 & 0 \\ - & - & 1 & 1 & 0 \\ - & - & 1 & 1 & 1 \end{bmatrix} \end{array} \tag{3.32}$$

with the following P matrix,

$$P = \begin{array}{c} \\ \mathrm{p}_0 \\ \mathrm{p}_1 \\ \mathrm{p}_5 \\ \mathrm{p}_3 \\ \mathrm{p}_4 \end{array} \begin{array}{c} \begin{array}{ccccc} [0^-] & [0^+] & [1^+] & [2^+] & [3^+] \end{array} \\ \begin{bmatrix} - & - & - & - & - \\ - & - & - & - & - \\ - & - & \frac{1}{2} & \frac{1}{2} & 0 \\ - & - & \frac{1}{2} & \frac{1}{2} & 0 \\ - & - & 0 & 0 & 1 \end{bmatrix} \end{array} \tag{3.33}$$

While storing each iteration of the W and P matrices, the loop is continued until results are converged. For the current W and P storage implementation, see Section 3.4.4.

One may ask if the worker and ensemble ratio $N_\mathrm{w} : N_\mathrm{ens}$ matters for overall performance, as ∞RETIS technically reduces to standard TIS when $N_\mathrm{w} = N_\mathrm{ens}$, and into a flavor of standard RETIS when $N_\mathrm{w} = 1$. Through an investigation using simple test systems, the optimal ratio appear to occur at around $1 : 2$. See paper A for the benchmark results and paper C for more realistic applications.

3.3.1 Interface Placement and Initialization

The maximum compute efficiency for TIS can be estimated to occur when the conditional crossing probability $P\left(\lambda_{i+1}|\lambda_i\right) \approx 0.2$ [29], with a slightly higher number (≈ 0.3) for RETIS. This thumb-rule number consequently estimates the ideal interface placement location and the total number N_{ens} to be initiated, as one would like both,

$$\begin{gathered} P\left(\lambda_{i+1}|\lambda_i\right)^{N_{\text{ens}}-1} \approx P\left(\lambda_B|\lambda_A\right) \\ P\left(\lambda_{i+1}|\lambda_i\right) \approx 0.2 \end{gathered} \tag{3.34}$$

For example, given the following $P\left(\lambda|\lambda_A\right)$ curve shown in Figure 3.3.2 with,

$$P\left(\lambda_B|\lambda_A\right) = 5.89 \cdot 10^{-7} \tag{3.35}$$

then one solution results in $N_{\text{ens}} = 10$ and $P\left(\lambda_{i+1}|\lambda_i\right) = 0.203$ as shown in Figure 3.3.2 **Left**. On the other hand, the ∞RETIS ensemble scalability with hardware could result in $P\left(\lambda_{i+1} \mid \lambda_i\right)$ values being much higher than 0.2. To keep a set of homogeneous $P\left(\lambda_{i+1} \mid \lambda_i\right)$ values, an alternative way of distributing interfaces for ∞RETIS could be to follow this simple equation,

$$P_A\left(\lambda_{i+1} \mid \lambda_i\right) \approx P_A\left(\lambda_B \mid \lambda_A\right)^{1/(N_{\text{ens}}-1)} \tag{3.36}$$

Resulting in $P\left(\lambda_{i+1}|\lambda_i\right) = 0.630$ with $N_{\text{ens}} = 32$ using the previous example (Figure 3.3.2 **Right**).

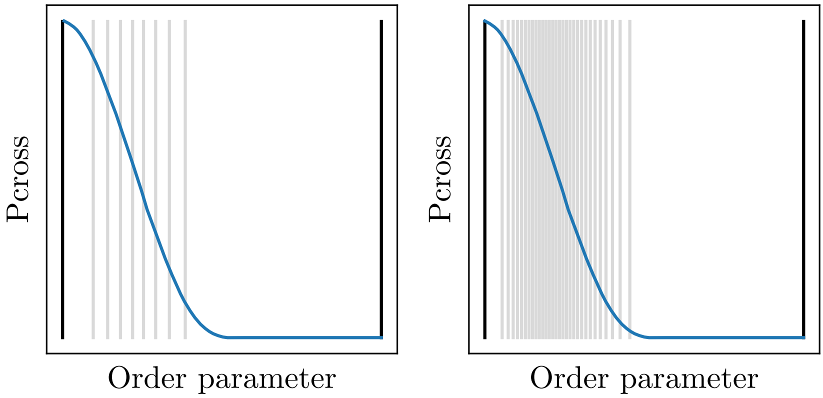


Figure 3.3.2: The (logarithmic) Pcross curve $P\left(\lambda|\lambda_A\right)$ with interfaces from a standard RETIS simulation (**Left**) and ∞RETIS (**Right**). The curve plotted is for a double-well system.

Notice, however, that a $P\left(\lambda|\lambda_A\right)$ curve is required to place interfaces at their ideal positions, which is a property that is generally obtained at the end of a simulation. Therefore, current practice is to perform an initial estimate of the interfaces and then iterate until satisfactory placements,

1. Make educated interface placement guesses (including states A and B).

2. Create valid trajectories for each ensemble.

3. Run a "short" ∞RETIS simulation to estimate $P_A\left(\lambda_{i+1} \mid \lambda_i\right)$. High variance and excessively high/low values could be a sign of bad placements, so faster simulation convergence could be to return to step 1 with the current information in mind. Otherwise, continue to a production run with current interfaces.

As step 1 requires the manual labor of an "educated guess", this current initialization scheme is not automated and can be quite arduous, especially if we need to keep creating valid load trajectories. A better method could possibly be to devise an algorithm that fluidly goes from automatic initialization to a production run within one simulation run. For example, the initialization phase could possibly take care of both generating initial trajectories and optimizing interface placements until some converge limit is reached, automatically switching the initialization phase to a production run. While devising such an algorithm is possible for RETIS, ∞RETIS would again benefit from the ability to initiate many workers in parallel to maximize throughput. Given that detailed balance is inconsequential during initialization, such a strategy has been devised on paper (video [67]) by Titus van Erp. Having such a smooth initialization phase in place would additionally alleviate the problem of non-specialists running path sampling simulations, as manual interface input and creating load trajectories would no longer be needed.

Note that in ∞RETIS, the bottle neck for maximum wall time convergence becomes the size of the maximum permanent block in the weight matrix W. Having more hardware available means that higher number of workers can be initiated (and hence N_{ens} can be increased), which results in larger W blocks as more paths will cross the next interface. Permanent calculations could therefore occupy the majority of the simulation time once the maximum W block is of a considerable size. So far, this upper ensemble number limit has not been reached when running the realistic ensembles listed in Paper C. However, even with many ensembles, having a very large block is generally improbable. In the case big blocks do happen, however, those permanents can be approximated by MC if needed.

3.4 ∞RETIS Software Implementation

The main ∞RETIS framework has been covered by the two previous chapters and papers A and C, but details regarding software implementation have been absent. Here I would like to highlight certain software matters that might not be obvious from pen and paper equations. The current work-in-progress code can be found in the following repository: `https://github.com/infretis/infretis`.

3.4.1 Scheduler and Workers: Dask Distributed

∞RETIS depends on a worker-scheduler functionally, a feature that the Python package Dask distributed (v2023.3.0) [68] provides out of the box. Specifically, the ∞RETIS *scheduler* should receive results from *workers* in the order of completion. If the scheduler has submitted certain tasks to be done to a number of workers, then the first worker to complete the task should also report back first. That way, the scheduler can immediately process the data (calculate W and P) and submit a new task to be done by the worker that just became available.

3.4.2 Hardware Handling

The ways to divide the allotted hardware to the number of initiated workers in an ∞RETIS simulation depends on various factors, like the MD program being run, if one or multiple nodes are used and whether or not simulations are run on high-performance computing (HPC) clusters. One additional factor to be considered is whether multiple GPU-acceleratable simulations can utilize the same GPU while running in parallel. For example, depending on the simulated system size, the total computing capability provided by a GPU might not be fully utilized by one MD simulation. By enabling NVIDIA Multi-Process Service (MPS) [50], multiple independent simulations can access compute resources provided by the same GPU and hence maximize the total GPU utilization possible. As a consequence, *ns/day* speedups from 1.3x (96K atoms) to 6.0x (6k atoms) can be achieved in comparison to running one single simulation accessing all of the GPU by itself [50] [paper C]. In paper C, multiple ∞RETIS simulations utilized MPS to initiate 10-16 workers while running on a single GPU-equipped node. To run 10 parallel workers using a GPU node with 20 CPU threads, the following GROMACS commands should be run 10 times

```
gmx mdrun -pin on -pinoffset X -pinstride 1 -ntomp 2 -ntmpi 1
```

with X being a multiple of 2 and the other settings are required to prevent multiple simulations utilizing the same CPU resources. Based on personal experience, however, the `pinstride` value (among other setttings) might have to be changed to obtain optimal performance. Even higher throughput could possibly have been achieved on the same GPU if the CPU provided 40 CPU threads instead of the 20, allowing each worker to utilize a shared GPU and 4 individual compute threads. Paper C also ran ∞RETIS on HPCs, accessing multiple GPU-equipped nodes per simulation. The HPCs employ a scheduler program called SLURM [69, 70] that provides a general way to allocate hardware to certain *subjobs* within one user-submitted job. One of the paper C simulations requested 20 GPU nodes for 20 workers within one SLURM job, with each of the 20 workers running the following command,

```
srun -n 1 -N 1 -G 1 --exact gmx mdrun
```

where `-n` is tasks, `-N` is nodes, `-G` is GPUs and `--exact` means no other tasks can access the same compute resources. In this case, the hardware allocation is bound to the SLURM utility `srun` and not to the specific MD command `gmx mdrun`. Given the capability of requesting multiple GPU nodes, however, higher throughput (*ns/day*) should be possible with MPS by for example initiating a total of 200 workers running 10 independent simulations per GPU node. Omitting details, however, enabling such a MPS setup within one SLURM job has simply put been unsuccessful for that HPC cluster accessed during this PhD. General user support for running multiple independent GPU-accelerated simulations on multiple nodes within one SLURM job is reasonably lacking. Any general user that requires such setups would most likely just divide the simulations into multiple SLURM jobs, something that is not currently possible for ∞RETIS since ensembles communicate with each other. Without MPS, however, running multiple CPU-based simulations on multiple nodes (within one SLURM job) is trivial, by simply setting the number of tasks (simulations) to be run in parallel per node in the SLURM job script,

```
-ntasks-per-node=X
```

3.4.3 Simulation Restart

Restarting a crashed MC or MD simulation is generally trivial, as one simply needs to continue from the last calculated step. Restarting ∞RETIS is also technically trivial, as one can simply just send the same number of workers to complete the simulations left undone by the crash. Obtaining identical results to a non-crashed simulation, however, can be non-trivial and possibly impossible. The reinstated workers might be handed alternative hardware that reshuffles the order of completion, causing different W and P matrices to be calculated. Stopping and restarting ∞RETIS simulations therefore introduce perturbations to the Markov chain. While the perturbation magnitude is currently unmeasured, the impact assumed is assumed to be relatively insignificant as long as long, uninterrupted simulations are run. In any case, the best practice would therefore be to minimize the amounts of restarts being performed.

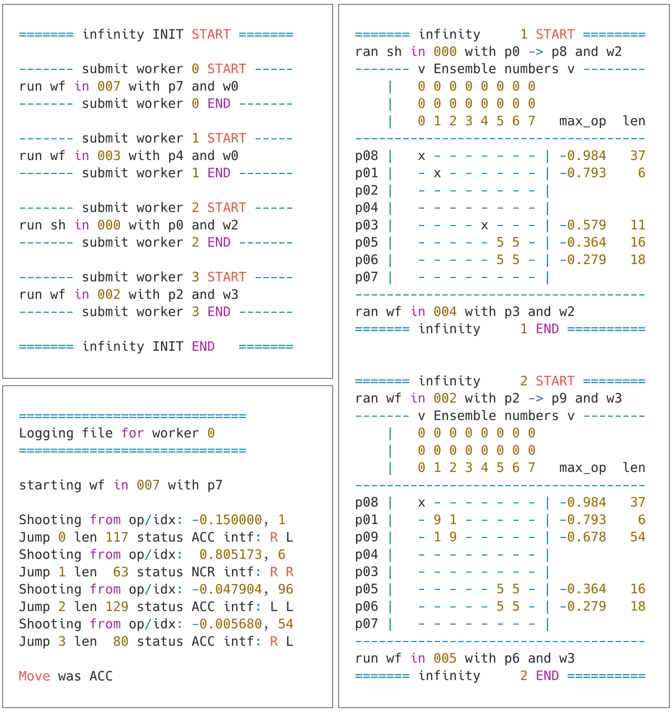
3.4.4 Simulation Output

The current simulation output from the ∞RETIS software has the same structure and flow as the ∞RETIS example in Section 3.3, shown in Figure 3.4.1 **Top Left** and **Right**. In addition to the general simulation output, each worker has an individual *worker.log* shown in Figure 3.4.1 **Bottom Left**.

A source of user confusion can be the path naming in ∞RETIS, $p_0, p_1, p_2, \cdots$, as seen in Figure 3.4.1. Since paths now no longer "belong" to individual ensembles due to infinite swapping, identifying paths to specific ensembles does not necessarily make sense. However, paths are regardless unique and should have an easy user-readable "path number" identifier, so new paths are marked as the Nth new produced path received by the scheduler, pN.

Another simple but practical consequence of ensemble independence is that all accepted paths can now simply be stored in a single folder, `traj/{path_number}` instead of in individual ensemble folders like `000/traj/{path_number}` as is done in PyRETIS.

In terms of calculating the rate, the path length, maximum order parameter, accumulative probabilities, and ensemble weights are stored in one single line in "`infretis_data.txt`" whenever a path is *pushed out*, see Figure 3.4.2. It is also because of the *pushed out* feature that the vertical path numbers in 3.4.2 and 3.4.1 **Right** are not necessarily listed in numerical order.

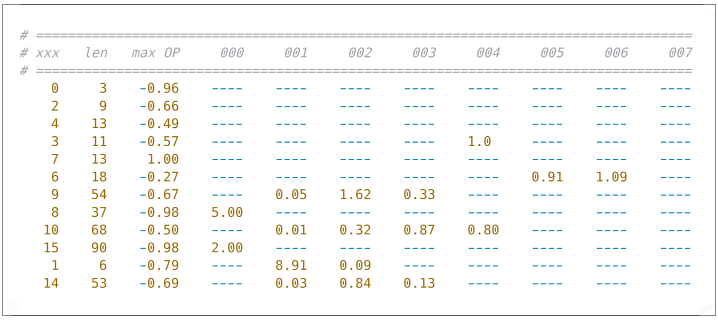


```
======= infinity INIT START =======

------- submit worker 0 START -----
run wf in 007 with p7 and w0
------- submit worker 0 END -------

------- submit worker 1 START -----
run wf in 003 with p4 and w0
------- submit worker 1 END -------

------- submit worker 2 START -----
run sh in 000 with p0 and w2
------- submit worker 2 END -------

------- submit worker 3 START -----
run wf in 002 with p2 and w3
------- submit worker 3 END -------

======= infinity INIT END   =======
```

```
============================
Logging file for worker 0
============================

starting wf in 007 with p7

Shooting from op/idx: -0.150000, 1
Jump 0 len 117 status ACC intf: R L
Shooting from op/idx:  0.805173, 6
Jump 1 len  63 status NCR intf: R R
Shooting from op/idx: -0.047904, 96
Jump 2 len 129 status ACC intf: L L
Shooting from op/idx: -0.005680, 54
Jump 3 len  80 status ACC intf: R L

Move was ACC
```

```
======= infinity     1 START ========
ran sh in 000 with p0 -> p8 and w2
------- v Ensemble numbers v --------
    |   0 0 0 0 0 0 0 0
    |   0 0 0 0 0 0 0 0
    |   0 1 2 3 4 5 6 7   max_op  len
-------------------------------------
p08 |   x - - - - - - - | -0.984   37
p01 |   - x - - - - - - | -0.793    6
p02 |   - - - - - - - - |
p04 |   - - - - - - - - |
p03 |   - - - - x - - - | -0.579   11
p05 |   - - - - - 5 5 - | -0.364   16
p06 |   - - - - - 5 5 - | -0.279   18
p07 |   - - - - - - - - |
-------------------------------------
ran wf in 004 with p3 and w2
======= infinity     1 END ==========


======= infinity     2 START ========
ran wf in 002 with p2 -> p9 and w3
------- v Ensemble numbers v --------
    |   0 0 0 0 0 0 0 0
    |   0 0 0 0 0 0 0 0
    |   0 1 2 3 4 5 6 7   max_op  len
-------------------------------------
p08 |   x - - - - - - - | -0.984   37
p01 |   - 9 1 - - - - - | -0.793    6
p09 |   - 1 9 - - - - - | -0.678   54
p04 |   - - - - - - - - |
p03 |   - - - - - - - - |
p05 |   - - - - - 5 5 - | -0.364   16
p06 |   - - - - - 5 5 - | -0.279   18
p07 |   - - - - - - - - |
-------------------------------------
run wf in 005 with p6 and w3
======= infinity     2 END ==========
```

Figure 3.4.1: Current (edited) output from the "pre-release" ∞RETIS software for a simulation with 4 workers and 8 ensembles. **Top Left:** The initialization phase. **Right:** The output every time a worker finishes a CP move. In the matrices, "x" denotes that the path 100% belongs to a certain ensemble. A number denotes fractional occupancy. A complete row or column of "-" denotes that the path and ensemble are busy. **Bottom Left:** The output by worker 0 performing a Wire Fencing move. Here 000 is $[0^-]$, 001 is $[0^+]$, 002 is $[1^+]$ and so on.



```
# ==============================================================================
# xxx    len    max OP      000      001      002      003      004      005      006      007
# ==============================================================================
    0      3    -0.96      ----     ----     ----     ----     ----     ----     ----     ----
    2      9    -0.66      ----     ----     ----     ----     ----     ----     ----     ----
    4     13    -0.49      ----     ----     ----     ----     ----     ----     ----     ----
    3     11    -0.57      ----     ----     ----     ----     1.0      ----     ----     ----
    7     13     1.00      ----     ----     ----     ----     ----     ----     ----     ----
    6     18    -0.27      ----     ----     ----     ----     ----     0.91     1.09     ----
    9     54    -0.67      ----     0.05     1.62     0.33     ----     ----     ----     ----
    8     37    -0.98      5.00     ----     ----     ----     ----     ----     ----     ----
   10     68    -0.50      ----     0.01     0.32     0.87     0.80     ----     ----     ----
   15     90    -0.98      2.00     ----     ----     ----     ----     ----     ----     ----
    1      6    -0.79      ----     8.91     0.09     ----     ----     ----     ----     ----
   14     53    -0.69      ----     0.03     0.84     0.13     ----     ----     ----     ----
```

Figure 3.4.2: The current "`infretis_data.txt`" file shows the path number, path length, maximum order parameter, and accumulative swapping fractions. The last 8 columns listing ensemble weight columns are not shown here. The first number of paths (rows) appear to have zero accumulative probabilities in any ensembles, as they are likely to have been immediately pushed out before any swap could have occurred. Path number 8 (among other paths) appears to only have integer weights in a single ensemble, which means that the path had no other valid ensembles/paths to swap with, which is logical for a path in $[0^-]$.

CHAPTER

FOUR

APPLICATION AND ANALYSIS

A number of applications have been studied using path sampling methods in this thesis, including the ruthenium redox reaction and thin film breakage described in paper B, and water dissociation, water boiling and the unfolding of the chignolin mini-protein described in paper C. Significant emphasis on data analysis has not been performed in those papers, however, as the main focus has been on method development. Additionally, many of these examples have previously been studied before [53, 71, 72, 73, 74]. In this chapter, we will perform preliminary analysis on the ruthenium redox reaction that was sampled using the WF move in Paper B, in addition to the chemical reaction of carbon dioxide converting into carbonic acid.

4.1 Electron Transfer Reactions

Electron transfer reactions lie at the core of reduction-oxidation reactions that occur within systems like batteries [75], and also play a central role in photochemical reactions involving the storage of solar energy [76] and within biological phenomena like photosynthesis [77]. One of the fundamental types of electron transfer reactions deals with the reduction-oxidation reaction between two metal ions solvated in water,

$$\mathbf{M}^{n+}_{(aq)} + \mathbf{M}'^{(n+1)+}_{(aq)} \rightarrow \mathbf{M}^{(n+1)+}_{(aq)} + \mathbf{M}'^{n+}_{(aq)} \tag{4.1}$$

where, described by Marcus theory [78, 79], the occurrence of electron transfer is largely influenced by the geometry and reorganization of the solvent. While Marcus theory is inherently macroscopic [80], its major predictions have been verified experimentally [81] and through running biased molecular dynamics simulations [82, 83]. To obtain insight into the microscopic details of solvent dynamics, unbiased MD simulations can be run to investigate collective variables related to the occurrence of reactions, like the patterns and networks formed by the aqueous hydrogen-bonded network. To actually observe reactions, however, may require the application of path sampling simulations as observing electron transfer reactions are inherently rare within MD simulations. Additionally, to avoid the difficult problem of connecting the configurational state of the solvent to the progress of the reaction, a qualitative TPS study has been performed [71] by running *ab initio*

MD and defining an order parameter that depends on transforming the resulting molecular orbitals into Maximally Localized Wannier Functions (MLWF) [84]. With the collaboration of Prof. Ensing of the TPS study [71], we initially applied RETIS to study the ruthenium self-exchange reaction, shown in Equation 4.2.

$$\mathrm{Ru}^{2+}_{(\mathrm{aq})} + \mathrm{Ru}'^{3+}_{(\mathrm{aq})} \rightarrow \mathrm{Ru}^{3+}_{(\mathrm{aq})} + \mathrm{Ru}'^{2+}_{(\mathrm{aq})} \tag{4.2}$$

With the establishment of the ∞RETIS protocol, however, we also applied it to study this same system. The reaction progress can be tracked as done by the TPS study, by following the MLWF orbital carrying the transferring electron from the electron donor to the electron acceptor. For the MD setup, two ruthenium ions are solvated in a periodic box of sides 12.4138 Å with 63 water molecules and a hydroxyl ion, resulting in a system with 193 atoms having a total surplus charge of 4+. The excess charge is neutralized through the introduction of a uniform background charge distribution from Ewald summation of long-range interactions. The system and molecular dynamics setup is also identical to the TPS study. Notably, the MD is propagated at 300 K using Density Functional Theory (DFT) where valence electrons are treated explicitly while core electrons are approximated by pseudopotentials. As with a surplus charge of 4+, 513 unrestricted MLWF orbitals thus result from the total number of ruthenium, oxygen and hydrogen valence electrons in the system. To sample this reaction using (∞)RETIS, a simple order parameter function can be defined based on the MLWF orbital centers,

$$\lambda_{\mathrm{ET}} = \frac{(d_{\mathrm{Ru}-X} - d_{\mathrm{Ru}'-X})}{d_{\mathrm{Ru}-\mathrm{Ru}'}} \tag{4.3}$$

where d is a distance function between two positions, Ru is the initial electron donor, Ru^{2+}, Ru′ is the initial electron acceptor, Ru^{3+}, and X is the transferring MLWF. The order parameter is illustrated in Figure 4.1.1.

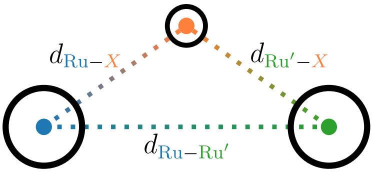


Figure 4.1.1: The two big balls represent the ruthenium ions and the small ball represent the transferring MLWF orbital. The dotted lines represent the distances used for the order parameter calculation in Equation 4.3.

X will initially reside within Ru, resulting in a reactant state value of $\lambda_{\mathrm{ET}} \approx -1$. After a successful electron transfer, $\lambda_{\mathrm{ET}} \approx 1$ identifies the product state. To identify X among all the 523 individual orbitals, a list of MLWFs and their closest oxygen or ruthenium neighbours can be constructed. If either of the two ruthenium ions has 6 closest MLWF neighbours, then the transferring orbital is simply the one farthest away from that ruthenium ion. Otherwise, both ruthenium ions have only 5 orbitals, so the transferring MLWF is one farthest away out of all the MLWFs connected to the O with one in excess. While simple, this allows for a robust characterization of the reaction progress.

During the initial sampling process, however, intermediate hydronium (H_3O^+) were observed to occasionally form. To avoid ending in the reactant or product state while having hydronium present, we define an altered order parameter λ_{HET},

$$\lambda_{\text{HET}} = \begin{cases} \lambda_B - 0.001 & \text{if } \lambda_{\text{ET}} > \lambda_B \text{ and hydronium is present} \\ \lambda_A + 0.001 & \text{if } \lambda_{\text{ET}} < \lambda_A \text{ and hydronium is present} \\ \lambda_{\text{ET}} & \text{otherwise} \end{cases} \tag{4.4}$$

that remains between $\lambda_A < \lambda_{\text{HET}} < \lambda_B$ when hydronium ion(s) are present. Otherwise, $\lambda_{\text{HET}} = \lambda_{\text{ET}}$.

An initial, reactive trajectory was generated by the authors of the initial TPS study, by running constrained MD. Given the TPS data, we performed a 238-day RETIS simulation where all but the zero ensembles utilized the WF move with high acceptance and with $P_{\text{RE}} = 0.5$. 100 CPUs were allocated to run the DFT MD via CP2K [85]. Later, when ∞RETIS was developed and hardware were available, the resulting RETIS P_{cross} curve was utilized to estimate the interface placements for a following 25-day ∞RETIS simulation with 20 workers and 41 ensembles (also with HA WF). Each worker was allocated 24 CPUs. Simulation results are shown in Table 4.1.1 and visualized in Figure 4.1.2.

Table 4.1.1: The rate, flux and crossing probability for a 238-day RETIS simulation and a 25-day ∞RETIS simulation are displayed.

Simulation	Rate [s^{-1}]	Flux [s^{-1}]	$P_A(\lambda_B\|\lambda_A)$
RETIS	$2.40 \times 10^9 \pm 146\%$	$1.692 \times 10^{13} \pm 2\%$	$1.421 \times 10^{-4} \pm 149.0\%$
∞RETIS	$6.32 \times 10^9 \pm 31\%$	$2.249 \times 10^{13} \pm 13.0\%$	$2.811 \times 10^{-4} \pm 28\%$

As can be seen in Figure 4.1.2 **Top Left** (and **Bottom Left**), more paths have been sampled by ∞RETIS while using a much shorter amount of wall time compared to RETIS. Additionally, as one would expect, more paths are sampled in the lower ensembles for ∞RETIS due to lower CP completion times. A deviation is seen in the initial point/ensemble, however, which is understandable as $[0^+]$ runs the shooting move without high acceptance. Consequently, "standard" acceptance yields higher rejectance probabilities and a higher number of rejected swaps between other ensembles. The cause for the latter effect is due to not sampling a "biased" path distribution that has a higher statistical average in maximum path order parameter values (see Section 3.1.1).

The P_{cross} curves in Figure 4.1.2 **Top Right** appear to correspond well between RETIS and ∞RETIS in the initial region between -1.0 and -0.75, where afterwards the ∞RETIS curve flattens and then sharply drops at 0.75 and onwards. The RETIS curve in comparison, appears to linearly decrease until arriving at state B. Apart from ∞RETIS to likely converge faster, a possible cause for the difference could lie in the sparseness of RETIS interfaces in comparison ∞RETIS, shown visually by the colored vertical dotted lines (and in Figure 4.1.2 **Top Left**). None of the RETIS ensemble interfaces λ_i lie between 0 and 1. Rather, only 2-3 ensembles sample trajectories that always start after the initial, steep electron donor region between -1.0 and -0.75, causing possible high uncertainties in that region. If we overlook the data discrepancy, however, the P_{cross} difference leads to

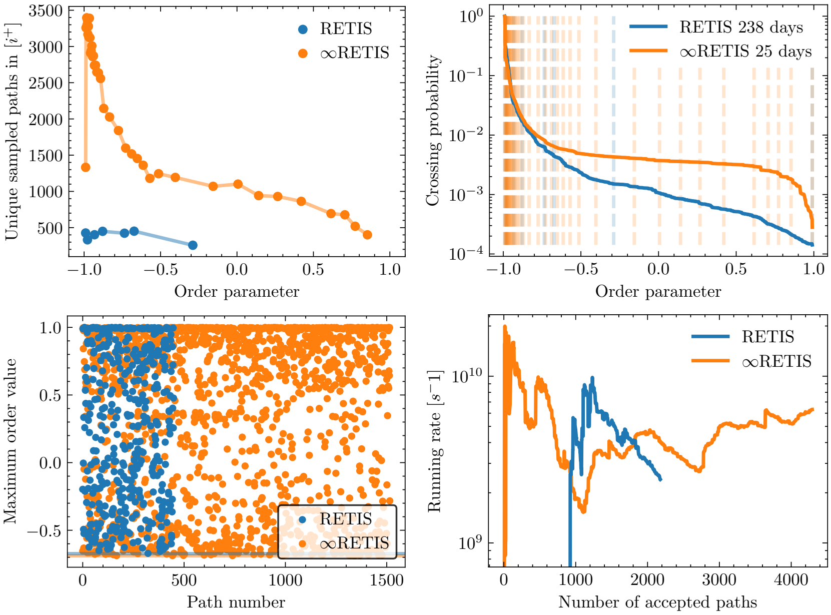


Figure 4.1.2: Top Left: The total number of unique (accepted) paths sampled for each ensemble, indicated by their λ_i placement on the order parameter x axis. Note 1, a path can be valid and counted in multiple ensembles due to being swapped around. Note 2, even if RETIS samples paths in cohort, rejectance in shooting/WF/RE lead to uneven number of unique sampled paths in each RETIS ensemble. **Top Right:** The crossing probability for the self-exchange reaction between Ruthenium ions. The respective RETIS and ∞RETIS interfaces are plotted as vertical dotted lines. **Bottom Left:** A scatter plot of the maximum order parameter values for all the paths sampled in two ∞RETIS and RETIS ensembles having similar λ_i values. Note that each path (scatter point) is counted in according to its specific accumulated probability p_{ij} and high acceptance weight w_{ij} values when calculating properties. **Bottom Right:** The running estimate of the rate against the number of accepted paths. The reason why the curve(s) starts after 0 can be due to having sampled zero reactive paths in the preceding samples.

different physical interpretations of the ensuing developments after the transferring electron escapes the initial electron donor. For ∞RETIS, a path that crosses -0.75 has a high probability to also cross 0.75 after the orbital initially "breaks free" from the electron donor. To completely arrive at the electron acceptor, however, appears to be moderately difficult as the crossing probability drops quite considerably close to $\lambda_{ET} = 1$. The cause for this behaviour could be due to the time difference that exists between transferring an electron and solvent reorganization [86]. Even if an electron has quickly "travelled" close to the electron acceptor, to remain there may require a reorganization in the solvent that does not always occur within a short time frame. This interpretation may also make more "sense" compared to the linear decrease in RETIS's P_{cross} curve. In both P_{cross} curves, however, there is an absence of a point of no return, a flat region leading towards state B. Therefore, even if "reactive" paths has been sampled, if the end point is further propagated, there still exist a probability for the path to quickly end up back in state A again, like if the solvent has not reorganized itself. So while the order parameter faithfully indicate the progress of an electron transfer, additional constraints / collective variables characterizing the solvent state may be required in order to define the true stable product state, see a simple sketch in Figure 4.1.3 **Left**.

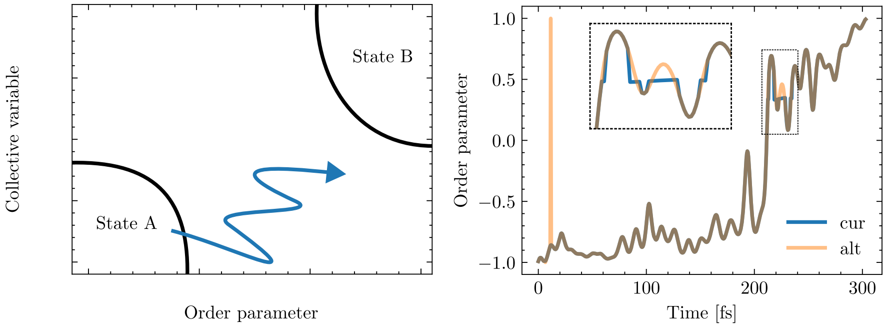


Figure 4.1.3: Left: Sketch of a possible reactive trajectory that reached state B in one axis (order parameter) but not in other, important axes (collective variables). **Right:** Current and alternative ways of selecting the travelling X. The order parameter values are calculated for a reactive electron transfer trajectory.

A specific issue or observation relating to the method of locating the transferring electron orbital X was detected when plotting the order parameter values of sampled trajectories, as seen in Figure 4.1.3 **Right**. The original way of selecting X, displayed in blue, does not follow one "specific" orbital from start to end, but selects X per MD frame based on neighbour lists and distances between O and Ru atoms. In the enclosed mini-figure, the blue curve form horizontal plateaus that are incongruent to the prior and latter oscillating behaviour. The cause for this behaviour appears to be due to X oscillating in close proximity to an O atom, resulting in the X selection method selecting an X farther away that does not represent the transferring electron orbital. While oscillating closely to an O atom may perhaps be questionable (or interesting), plateaus can be observed in a number of the sampled trajectories. An alternative selection scheme may be designed by simply following the index of the orbital leaving the electron donor, shown by the pale orange curve that does not form plateaus. With this scheme, however, a different issue can appear. At least for when the simulations were run (2021-2023), there was no guarantee that the index list for the MLWFs would remain the same throughout a MD trajectory. For example, the index list could possibly change, as appeared to have happened by the early, light orange peak in Figure 4.1.3 **Right**. A snapshot of the reactive trajectory shown in in Figure 4.1.3 **Right** is displayed in Figure 4.1.4.

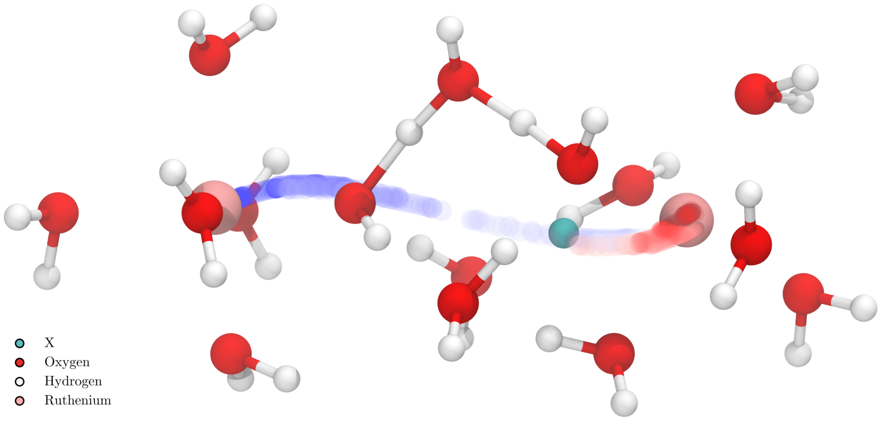


Figure 4.1.4: A snapshot describing the transfer of X (the transferring electron orbital) from the donor to the acceptor, with a Grotthuss mechanism between two water molecules and one hydroxyl ion occurring at the same time. The blue-red curve is the superimposed path of X (the orange curve in Figure 4.1.3, excluding the spike), from all frames displayed together. Figure is made using VMD [87].

A recurrent characteristic of reactive sampled trajectories is the occurrence of proton transfer reactions during the electron transfer process. In the reactant state (or in the product state for that matter), the sole hydroxyl ion is part of of the initial electron acceptor's first solvation shell. As the electron transfers to the initial electron acceptor, so does (usually) two successive proton transfer reactions also occur, which leads to the formation of a hydroxyl ion close to the initial electron donor at the end, as can be envisioned in Figure 4.1.4. The process of transferring one proton to another via water molecules is called the Grotthuss mechanism. While the hydroxyl ion changed sides in the majority of the reactive trajectories, a number of the trajectories only observed the transfer of an electron. Likely, either the hydroxyl ion or the excess electron will quickly transfer if further propagated. Extensive chemical knowledge can possibly be obtained from the large amount of data generated from the RETIS and ∞RETIS simulations, by using algorithms like chemistrees [88] or predictive power [89]. Due to time, however, we will end this subchapter by plotting one simple collective variable, the average distance between a ruthenium atom and the oxygen atoms of the first solvent shell over time, as shown for two trajectories in Figure 4.1.5. The difference in equilibrium distance between the two ruthenium ions is likely due to the difference in acceptor and donor charge. While the trajectory's solvent shell distances oscillate between and under the equilibrium distance, the two Ru and Ru$'$ curves appear to be relatively inversely related with start and end values becoming flipped and being close to the equilibrium average. They also appear to be somewhat correlated with the order parameter value, as the solvent shell distances should partly be determined by the ruthenium charges and surrounding solvent molecules. Whether the transfer of the hydroxyl ion is correlated to the order parameter or other collective variables remains to be seen, however.

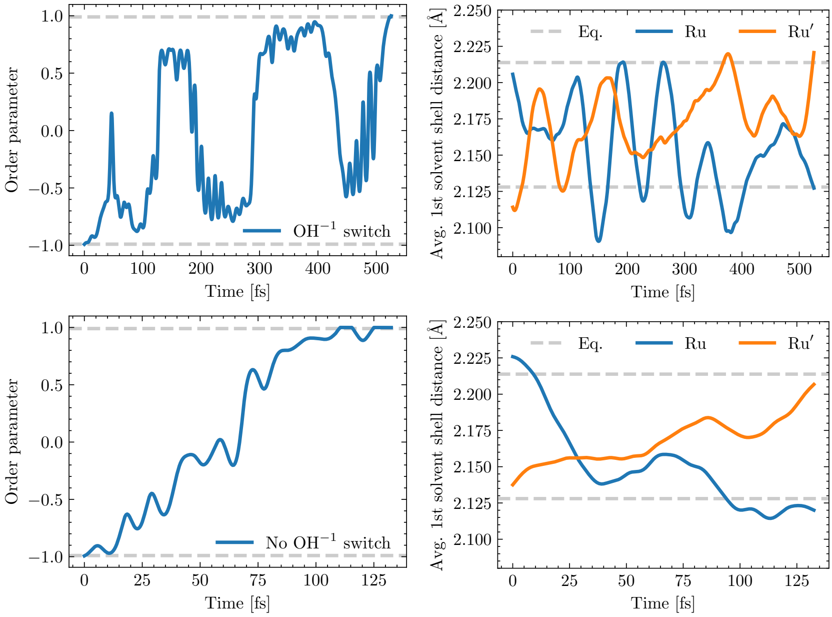


Figure 4.1.5: Left: The order parameter values of two reactive trajectories. **Right:** For the same trajectories, the average distance between the 6 inner-most oxygen atoms and Ru/Ru′ is plotted per frame. Equilibrium averages are also plotted as horizontal dotted grey lines.

4.2 Solvated Carbon Dioxide to Carbonic Acid

Solvated carbon dioxide (CO_2) chemistry play a key role in major, ongoing challenges like carbon capture for the prevention of atmospheric CO_2 release [90] and the acidification of the ocean caused by the formation of carbonic acid (H_2CO_3) from solvated CO_2 in seawater [91]. Considerable attention has therefore been dedicated to develop effective amine-based solvents for CO_2 capture [92], and more recently ways to remove solvated CO_2 from seawater [93]. While the two challenges may not appear to be directly related, bicarbonate (HCO_3^-) is the most common product observed in amine solutions [92], which is the same product that is obtained from the deprotonation of carbonic acid in seawater,

$$H_2CO_{3(aq)} \rightarrow HCO^-_{3(aq)} + H^+_{(aq)} \tag{4.5}$$

Other, catalyzed chemical reactions related to solvated CO_2 include the formation of formic acid [94, 95], carbon monoxide [96, 97] and alcohols like methanol [96] and ethanol [98]. Further study, through the application of sampling methods like ∞RETIS would therefore be highly beneficial for providing insight into the chemical and catalytic processes involving the reduction of CO_2 into other species. As a first step, we can apply ∞RETIS to study the CO_2 reaction forming carbonic acid in water,

$$CO_{2(aq)} + H_2O_{(aq)} \rightarrow H_2CO_{3(aq)} \tag{4.6}$$

which occurs without the need of catalysts at various experimental conditions, like in carbonated drinks and in the ocean. Similarly to the study of the water dissociation reaction in Paper C and the electron transfer reaction in the previous section, this reaction can be modelled using *ab initio* DFT dynamics. To determine the progress of this reaction, a simple order parameter can be the minimum angle between the carbon atom and two connected oxygen atoms. Once arriving at the product state, however, possibly three oxygen atoms will be bonded with the carbon atom. We therefore impose a pre-defined cut-off value of 1.41 Å. If three oxygen atoms exist within this radius, then the minimum out of the three O-C-O angles becomes the order parameter angle. We define the reactant state to be larger 173° degrees and the product state to be smaller than 114°. We additionally enforce the criteria of pure water at the product state, in addition to C having three O neighbors within the cutoff radius. The former criteria confine the product state to purely carbonic acid, as the product state might otherwise be a combination of carbonic acid and bicarbonate (deprotoned carbonic acid). MD snapshots are shown in Figure 4.2.1,

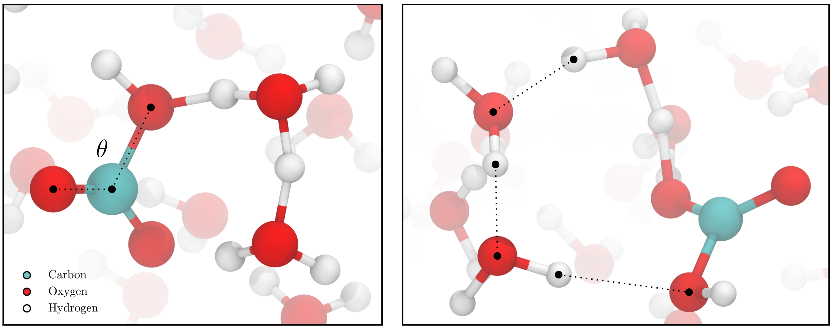


Figure 4.2.1: Snapshots of two reactive trajectories, with **Left** forming the *cis-trans* and **Right** forming the *cis-cis* carbonic acid conformer. **Left** illustrates the order parameter, and shows the beginnings of a water molecule nucleophillically attacking the carbon atom. **Right** illustrates the ending of a Grotthuss chain, with dotted lines showing the initial O-H pairs.

The MD was run with DFT MD (CP2K) at 300K, a cubic box with length 12.4138 Å and a system size of 64 water molecules and one carbon dioxide molecule. The DFT functional used was revPBE-D3 [99]. An initial reactive trajectory was speedily obtained by running a metadynamics simulation and having the O-C-O angle as the sole collective variable. Similarly, after determining interfaces from initial explorative ∞RETIS runs, a ∼2-day production simulation was run with 20 workers and 41 ensembles, with each worker utilizing 12 CPUs each. The crossing probability and the running calculation of the rate constant are shown in Figure 4.2.2. A point of no return can be observed in Figure 4.2.2 **Left**, where once $\theta \sim 130$ is crossed, the system is almost guaranteed to arrive at the product state. The calculated rate constant appears to converge towards a value that is two order of magnitudes away from experimentally reported values, shown in Figure 4.2.2 **Right**, which is quite impressive given the limited system size and simulation time. A better match could be obtained by increasing the two factors, and modifying the MD settings, like changing the DFT functional, basis sets and DFT convergence limits. The order parameter values of three paths are shown in Figure 4.2.3 **Left**,

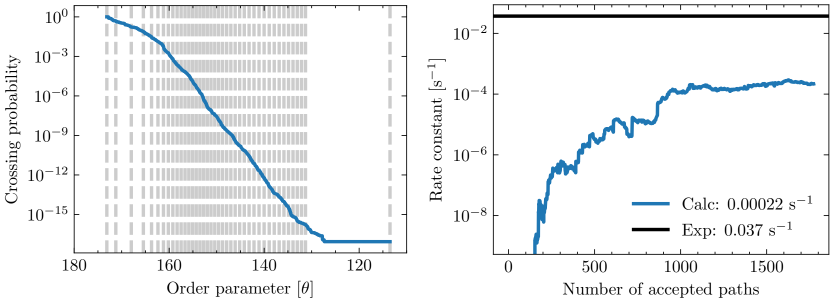


Figure 4.2.2: Left: The crossing probability of the reaction, with the interfaces plotted as vertical dotted lines. A plateau is formed around $\theta \sim 130$. **Right**: The running estimate of the rate against the number of accepted paths. The calculated rate constant from the ∞RETIS simulation equals 0.00022 $s^{-1} \pm 68\%$, with the flux and P_{cross} being $2.508{\times}10^{13} \pm 19\%$ s^{-1} and 8.609 $\times 10^{-18} \pm 62\%$ respectively. The plotted experimental value of 0.037 s^{-1} is from S. R. Emerson and J. I. Hedges [100] at 25 °C. Two other experimentally similar values were reported to be 0.039 s^{-1} by ChemEurope [25 °C] [101] and 0.0371 s^{-1} by A. L. Soli and R. H. Byrne also at 25 °C.

The primary observed carbonic acid formation mechanism occurs as illustrated in Figure 4.2.1 **Left**, where 19 out of the 20 sampled reactive trajectories resulted in the *cis-trans* carbonic acid conformer. Only one reactive trajectory formed the *cis-cis* conformer as shown in Figure 4.2.1 **Right**. In both cases, a water molecule's oxygen atom appears to release a proton and concurrently attack the carbon atom. The released proton then initiates the Grotthuss mechanism to deliver a proton to one of the other carbon-bonded oxygen atoms. One possible reason for the 19:1 conformer disparity is a difference in the required path length for the Grotthuss mechanism. As can be seen in Figure 4.2.1 **Left**, only two intermediate water molecules were generally required to form the *cis-trans* conformer. The sole *cis-cis* trajectory shown in Figure 4.2.1 **Right**, however, required three intermediate water molecules, and a longer time to complete than most of the other reactive paths, as seen in Figure 4.2.3 **Right**.

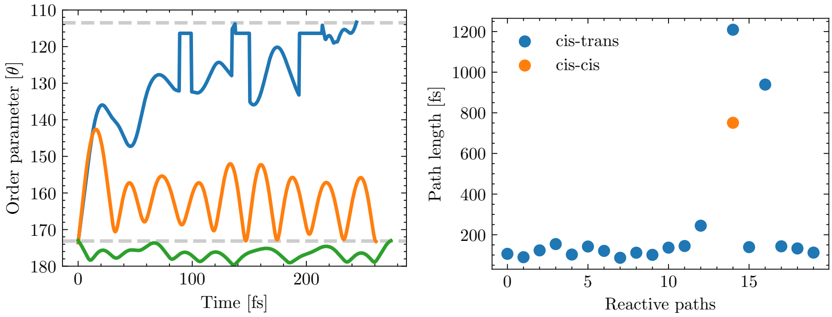


Figure 4.2.3: Left: The depiction of a reactive (blue) a unreactive (orange) and a [0^{-}] (green) path. **Right**: The length of sampled reactive paths, with one of them resulting in a *cis-cis* carbonic acid.

In all three paths that exceed 500 fs, three or more hydronium ion intermediates were formed throughout the reaction. In comparison, the shorter paths had only time to form two hydronium ions before arriving at the *cis-trans* product state. Therefore, as the *cis-cis* conformer potentially requiring longer time to form, the chance of falling back towards the reactant state possibly also increases, resulting in the observed conformer disparity. In fact, this result appears to well agree with a theoretical paper published in 1997 [102], that predicted the formation of carbonic acid via a "water chain mechanism" primarily involving the active participation of two water molecules (excluding the attacking one). Additionally, the illustrated geometries in the paper mainly show the formation of *cis-trans* conformers. If the two conformers remain relatively stable in solution, then the conformer formation disparity would also imply that most of the carbonic acid would exist as *cis-trans* conformer in water solution. The experimental findings of [103] possibly support this claim, as they managed to (only) detect the *cis-trans* conformer via "a supersonic jet using a pulsed discharge nozzle" and Fourier-transform microwave spectroscopy. For further work, more analysis and simulation should be performed. For example, the conformerization reaction between *cis-trans* and *cis-cis*, the stability of carbonic acid, and how various factors like box size, DFT functionals and simulation time would affect the prediction of rate constants.

CHAPTER FIVE

CONCLUSION AND OUTLOOK

Through a combination of various methods and enhancements, a method called ∞RETIS has been formulated to efficiently and quickly tackle the sampling of rare events in MD simulations. From proof of concept papers A and B, a coded-up ∞RETIS software is now available to quickly converge many realistic systems, like the study of water boiling, water dissociation, protein unfolding (paper C), redox reactions (Chapter 4.1) and carbonic acid formation (Chapter 4.2). Future ∞RETIS applications and developments will therefore be interesting to follow, especially as they can make use of the excellent performance, parallelizability, and accuracy provided by the algorithm. For example, ∞RETIS appears to be quite compatible with current hardware, software and algorithmic advancements:

- The hardware available can be effectively utilized. For example, GPU and CPU throughput can be maximized via MPS, and, for better or worse, storage IO and RAM usage increases linearly per parallel worker. ∞RETIS simulations can therefore expect to become faster with seemingly more and faster hardware options by the day.

- Different parallel simulations can run different types of MD. For example, various levels of theory could be run between the [0^-] and [i^+] ensembles, which is the idea for Quantis [29]. One configuration would be to run force field-based MD in [i^-], and let [i^+] run quantum mechanics/molecular mechanics [104] and/or pure quantum-based MD.

- As MLFF MD enables quantum accuracy with reduced costs, system sizes that previously were not possible can now be studied. To do so, however, require access to quality data, especially when describing transitionary events. Running ∞RETIS with MLFF would directly provide quality data in addition to sampling rare events at the same time.

Although ∞RETIS is highly adaptable and the major problem of wall time convergence has been solved through parallelization, certain other problems still hinder widespread applicability. A considerable issue is the fact that the term *rare events* is not necessarily well defined, so ∞RETIS cannot straightforwardly be applied to sample any type of rare event. Noticeably, a simulation will quickly break down in the presence of meta-stable states between the more stable reactant and product states, as shown in Figure 5.0.1,

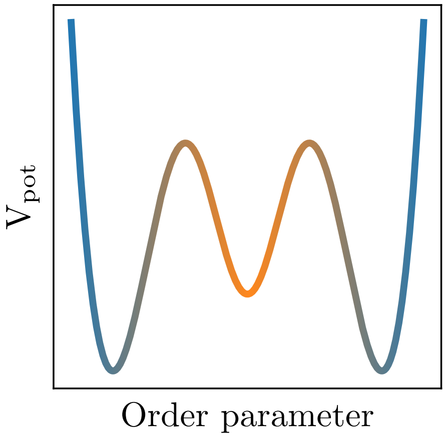


Figure 5.0.1: The potential energy curve of a system with two global minima and one local minima in the center. The orange area marks the metastable region.

In these systems, a launched shooting move might end up propagating forever within the orange region (in Figure 5.0.1). Additionally, areas where rare even sampling could be useful, like drug discovery [105] or catalysis optimization [106], often require the study of complicated systems that contain a plethora of metastable states. One example is the ∞RETIS study of chignolin protein unfolding (paper C), an artificial mini-protein [107] solvated in water. While the ∞RETIS simulations managed to converge and predict accurate results, many excessively long paths were also generated from the existence of a metastable state between states A and B.

In the case of simulating more complicated systems than the mini-protein, however, the effectivity of ∞RETIS simulations may dramatically suffer. To resolve this issue, one can perhaps use the lessons in this thesis, that problems can be resolved through the use of even more enhancements and algorithms. For example, instead of only defining two stable states, A and B, ∞RETIS could possibly be extended to cover multiple states (MS) [32], and/or to utilize shorter path definitions, like partial paths (PP) [34]. Other problems, like the current (arduous) initialization phase and interface placement optimization, also seem solvable (Section 3.3.1). Therefore, having already enabled a new, wide region of studiable systems, additional complications may be solved with more algorithmic enhancements. We therefore expect ∞RETIS, and its possible siblings (∞MSRETIS, ∞REPPTIS), to generate accurate results that help solve many of the current real-world issues we are dealing with today.

REFERENCES

[1] D. Frenkel and B. Smit. *Understanding Molecular Simulation: From Algorithms to Applications.* 2nd edition. San Diego: Academic Press, Nov. 2001. ISBN: 978-0-12-267351-1.

[2] M. E. Tuckerman. *Statistical Mechanics: Theory and Molecular Simulation.* 1st edition. Oxford ; New York: Oxford University Press, Apr. 2010. ISBN: 978-0-19-852526-4.

[3] N. Metropolis and S. Ulam. "The Monte Carlo method". In: *Journal of the American Statistical Association* 44.247 (Sept. 1949), pp. 335–341. ISSN: 0162-1459.

[4] L. Badger. "Lazzarini's Lucky Approximation of pi". In: *Mathematics Magazine* (Apr. 1994). ISSN: 0025-570X.

[5] B. J. Alder and T. E. Wainwright. "Phase Transition for a Hard Sphere System". In: *The Journal of Chemical Physics* 27.5 (Nov. 1957), pp. 1208–1209. ISSN: 0021-9606.

[6] V. Bråten et al. "Equation of state for confined fluids". In: *The Journal of Chemical Physics* 156.24 (June 2022), p. 244504. ISSN: 0021-9606.

[7] C. Pedebos and S. Khalid. "Simulations of the spike: molecular dynamics and SARS-CoV-2". In: *Nature Reviews Microbiology* 20.4 (Apr. 2022), pp. 192–192. ISSN: 1740-1534.

[8] T. Joutsuka and K. Ando. "Hydration Structure in Dilute Hydrofluoric Acid". In: *The Journal of Physical Chemistry A* 115.5 (Feb. 2011), pp. 671–677. ISSN: 1089-5639.

[9] A. C. T. van Duin et al. "ReaxFF: A Reactive Force Field for Hydrocarbons". In: *The Journal of Physical Chemistry A* 105.41 (Oct. 2001), pp. 9396–9409. ISSN: 1089-5639.

[10] T. P. Senftle et al. "The ReaxFF reactive force-field: development, applications and future directions". In: *npj Computational Materials* 2.1 (Mar. 2016), pp. 1–14. ISSN: 2057-3960.

[11] A. Pribram-Jones, D. A. Gross, and K. Burke. "DFT: A Theory Full of Holes?" In: *Annual Review of Physical Chemistry* 66.1 (2015), pp. 283–304.

[12] M. Zheng, X. Li, and L. Guo. "Algorithms of GPU-enabled reactive force field (ReaxFF) molecular dynamics". In: *Journal of Molecular Graphics and Modelling* 41 (Apr. 2013), pp. 1–11. ISSN: 1093-3263.

[13] A. Leach. *Molecular Modelling: Principles and Applications.* 2nd edition. Harlow, England ; New York: Pearson, Jan. 2001. ISBN: 978-0-582-38210-7.

[14] C. J. Geyer and E. A. Thompson. "Annealing Markov Chain Monte Carlo with Applications to Ancestral Inference". In: *Journal of the American Statistical Association* (Sept. 1995).

[15] W. D. Vousden, W. M. Farr, and I. Mandel. "Dynamic temperature selection for parallel tempering in Markov chain Monte Carlo simulations". In: *Monthly Notices of the Royal Astronomical Society* 455.2 (Jan. 2016), pp. 1919–1937. ISSN: 0035-8711.

[16] N. Plattner et al. "An infinite swapping approach to the rare-event sampling problem". In: *The Journal of Chemical Physics* 135.13 (Oct. 2011), p. 134111. ISSN: 0021-9606.

[17] N. Plattner, J. D. Doll, and M. Meuwly. "Overcoming the Rare Event Sampling Problem in Biological Systems with Infinite Swapping". In: *Journal of Chemical Theory and Computation* 9.9 (Sept. 2013), pp. 4215–4224. ISSN: 1549-9618.

[18] S. Duane et al. "Hybrid Monte Carlo". In: *Physics Letters B* 195.2 (Sept. 1987), pp. 216–222. ISSN: 0370-2693.

[19] M. S. Friedrichs et al. "Accelerating Molecular Dynamic Simulation on Graphics Processing Units". In: *Journal of computational chemistry* 30.6 (Apr. 2009), pp. 864–872. ISSN: 0192-8651.

[20] M. Själander et al. *EPIC: An Energy-Efficient, High-Performance GPGPU Computing Research Infrastructure.* Feb. 2022. URL: `http://arxiv.org/abs/1912.05848` (visited on 05/09/2024).

[21] C. D. Montgomery. "Factors Affecting Energy Barriers for Pyramidal Inversion in Amines and Phosphines: A Computational Chemistry Lab Exercise". In: *Journal of Chemical Education* 90.5 (May 2013), pp. 661–664. ISSN: 0021-9584.

[22] J. Kästner. "Umbrella sampling". In: *WIREs Computational Molecular Science* 1.6 (2011), pp. 932–942. ISSN: 1759-0884.

[23] A. Laio and M. Parrinello. "Escaping free-energy minima". In: *Proceedings of the National Academy of Sciences of the United States of America* 99.20 (Oct. 2002), pp. 12562–12566. ISSN: 0027-8424.

[24] C. Dellago et al. "Transition path sampling and the calculation of rate constants". In: *The Journal of Chemical Physics* 108.5 (Feb. 1998), pp. 1964–1977. ISSN: 0021-9606.

[25] P. G. Bolhuis et al. "TRANSITION PATH SAMPLING: Throwing Ropes Over Rough Mountain Passes, in the Dark". In: *Annual Review of Physical Chemistry* 53.1 (2002), pp. 291–318.

[26] T. S. van Erp, D. Moroni, and P. G. Bolhuis. “A novel path sampling method for the calculation of rate constants”. In: *The Journal of Chemical Physics* 118.17 (May 2003), pp. 7762–7774. ISSN: 0021-9606.

[27] T. S. van Erp and P. G. Bolhuis. “Elaborating transition interface sampling methods”. In: *Journal of Computational Physics* 205.1 (May 2005), pp. 157–181. ISSN: 0021-9991.

[28] T. S. Van Erp. “Reaction Rate Calculation by Parallel Path Swapping”. In: *Physical Review Letters* 98.26 (June 2007), p. 268301. ISSN: 0031-9007, 1079-7114.

[29] R. Cabriolu et al. “Foundations and latest advances in replica exchange transition interface sampling”. In: *The Journal of Chemical Physics* 147.15 (Oct. 2017), p. 152722. ISSN: 0021-9606.

[30] T. S. v. Erp. “How far can we stretch the timescale with RETIS?” In: *EPL* 143.3 (Aug. 2023), p. 30001. ISSN: 0295-5075, 1286-4854.

[31] L. R. Pratt. “A statistical method for identifying transition states in high dimensional problems”. In: *The Journal of Chemical Physics* 85.9 (Nov. 1986), pp. 5045–5048. ISSN: 0021-9606.

[32] J. Rogal and P. G. Bolhuis. “Multiple state transition path sampling”. In: *The Journal of Chemical Physics* 129.22 (Dec. 2008), p. 224107. ISSN: 0021-9606.

[33] D. Moroni, P. Bolhuis, and T. Van Erp. “Rate constants for diffusive processes by partial path sampling”. In: *Journal of Chemical Physics* 120.9 (2004), pp. 4055–4065. ISSN: 0021-9606.

[34] W. Vervust et al. “Path sampling with memory reduction and replica exchange to reach long permeation timescales”. In: *Biophysical Journal* 122.14 (July 2023), pp. 2960–2972. ISSN: 1542-0086.

[35] D. T. Zhang, E. Riccardi, and T. S. van Erp. “Enhanced path sampling using subtrajectory Monte Carlo moves”. In: *The Journal of Chemical Physics* 158.2 (Jan. 2023), p. 024113. ISSN: 0021-9606.

[36] S. Roet, D. T. Zhang, and T. S. van Erp. “Exchanging Replicas with Unequal Cost, Infinitely and Permanently”. In: *The Journal of Physical Chemistry A* 126.47 (Dec. 2022), pp. 8878–8886. ISSN: 1089-5639.

[37] D. T. Zhang et al. “Highly parallelizable path sampling with minimal rejections using asynchronous replica exchange and infinite swaps”. In: *Proceedings of the National Academy of Sciences* 121.7 (Feb. 2024), e2318731121.

[38] W. Vervust et al. “PyRETIS 3: Conquering rare and slow events without boundaries”. In: *Journal of Computational Chemistry* 45.15 (2024), pp. 1224–1234. ISSN: 1096-987X.

[39] M. P. Allen and D. J. Tildesley. *Computer Simulation of Liquids*. 2nd edition. Oxford: Oxford University Press, Aug. 2017. ISBN: 978-0-19-880320-1.

[40] R. Swendsen and J.-S. Wang. “Replica Monte Carlo Simulation of Spin-Glasses”. In: *Physical review letters* 57 (Dec. 1986), pp. 2607–2609.

[41] C. Dellago, P. Bolhuis, and P. Geissler. “Transition Path Sampling Methods”. In: *Computer Simulations in Condensed Matter Systems: From Materials to Chemical Biology Volume 1*. Lecture Notes in Physics. Berlin, Heidelberg: Springer, 2006, pp. 349–391. ISBN: 978-3-540-35273-0. URL: `https://doi.org/10.1007/3-540-35273-2_10` (visited on 01/29/2024).

[42] C. Liu. *The Three-Body Problem.* Trans. by K. Liu. New York: Tor Books, Jan. 2016. ISBN: 978-0-7653-8203-0.

[43] N. C. Stone and N. W. C. Leigh. “A statistical solution to the chaotic, non-hierarchical three-body problem”. In: *Nature* 576.7787 (Dec. 2019), pp. 406–410. ISSN: 1476-4687.

[44] G. W. Richings and S. Habershon. “Predicting Molecular Photochemistry Using Machine-Learning-Enhanced Quantum Dynamics Simulations”. In: *Accounts of Chemical Research* 55.2 (Jan. 2022), pp. 209–220. ISSN: 0001-4842.

[45] H. Li et al. “Light-Induced Ultrafast Molecular Dynamics: From Photochemistry to Optochemistry”. In: *The Journal of Physical Chemistry Letters* 13.25 (June 2022), pp. 5881–5893.

[46] S. D. Folkestad et al. “eT 1.0: An open source electronic structure program with emphasis on coupled cluster and multilevel methods”. In: *The Journal of Chemical Physics* 152.18 (May 2020), p. 184103. ISSN: 0021-9606.

[47] H. Jung et al. “Machine-guided path sampling to discover mechanisms of molecular self-organization”. In: *Nature Computational Science* 3.4 (Apr. 2023), pp. 334–345. ISSN: 2662-8457.

[48] M. J. Abraham et al. “GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers”. In: *SoftwareX* 1-2 (Sept. 2015), pp. 19–25. ISSN: 2352-7110.

[49] A. P. Thompson et al. “LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales”. In: *Computer Physics Communications* 271 (Feb. 2022), p. 108171. ISSN: 0010-4655.

[50] *Maximizing GROMACS Throughput with Multiple Simulations per GPU Using MPS and MIG.* Oct. 2021. URL: `https://developer.nvidia.com/blog/maximizing-gromacs-throughput-with-multiple-simulations-per-gpu-using-mps-and-mig/` (visited on 01/11/2024).

[51] O. T. Unke et al. “Machine Learning Force Fields”. In: *Chemical Reviews* 121.16 (Aug. 2021), pp. 10142–10186. ISSN: 0009-2665.

[52] H. E. Sauceda et al. “Molecular force fields with gradient-domain machine learning: Construction and application to dynamics of small molecules with coupled cluster forces”. In: *The Journal of Chemical Physics* 150.11 (Mar. 2019), p. 114102. ISSN: 0021-9606.

[53] D. Zahn. “How Does Water Boil?” In: *Physical Review Letters* 93.22 (Nov. 2004), p. 227801.

[54] K. Karalis et al. “Deciphering the molecular mechanism of water boiling at heterogeneous interfaces”. In: *Scientific Reports* 11.1 (Oct. 2021), p. 19858. ISSN: 2045-2322.

[55] N. Metropolis et al. “Equation of State Calculations by Fast Computing Machines”. In: *The Journal of Chemical Physics* 21.6 (Dec. 2004), pp. 1087–1092. ISSN: 0021-9606.

[56] A. Atashpendar, T. Schilling, and T. Voigtmann. “Sequencing chess”. In: *Europhysics Letters* 116.1 (Nov. 2016), p. 10009. ISSN: 0295-5075.

[57] E. Riccardi, O. Dahlen, and T. S. van Erp. “Fast Decorrelating Monte Carlo Moves for Efficient Path Sampling”. In: *The Journal of Physical Chemistry Letters* 8.18 (Sept. 2017), pp. 4456–4460.

[58] T.-Q. Yu et al. “Multiscale implementation of infinite-swap replica exchange molecular dynamics”. In: *Proceedings of the National Academy of Sciences* 113.42 (Oct. 2016), pp. 11744–11749.

[59] B. W. Zhang et al. “Simulating Replica Exchange: Markov State Models, Proposal Schemes, and the Infinite Swapping Limit”. In: *The Journal of Physical Chemistry B* 120.33 (Aug. 2016), pp. 8289–8301. ISSN: 1520-6106.

[60] J. Lu and E. Vanden-Eijnden. “Methodological and Computational Aspects of Parallel Tempering Methods in the Infinite Swapping Limit”. In: *Journal of Statistical Physics* 174.3 (Feb. 2019), pp. 715–733. ISSN: 1572-9613.

[61] D. G. Glynn. “The permanent of a square matrix”. In: *European Journal of Combinatorics* 31.7 (Oct. 2010), pp. 1887–1891. ISSN: 0195-6698.

[62] K. Balasubramanian. “Combinatorics and Diagonals of Matrices”. PhD thesis. Madras, India: Loyola College, 1980.

[63] E. Bax. “Finite-difference Algorithms for Counting Problems”. PhD thesis. Pasadena, United States of America: California Institute of Technology, 1998.

[64] A. Lervik, E. Riccardi, and T. S. van Erp. “PyRETIS: A well-done, medium-sized python library for rare events”. In: *Journal of Computational Chemistry* 38.28 (2017), pp. 2439–2451. ISSN: 1096-987X.

[65] E. Riccardi et al. “PyRETIS 2: An improbability drive for rare events”. In: *Journal of Computational Chemistry* 41.4 (2020), pp. 370–377. ISSN: 1096-987X.

[66] D. W. H. Swenson et al. “OpenPathSampling: A Python Framework for Path Sampling Simulations. 1. Basics”. In: *Journal of Chemical Theory and Computation* 15.2 (Feb. 2019), pp. 813–836. ISSN: 1549-9618.

[67] Titus van Erp. *Infinity RETIS 04: General strategy for initialisation. Input parameters.* Feb. 2023. URL: https://www.youtube.com/watch?v=wThZpYmYogY (visited on 01/11/2024).

[68] M. Rocklin. “Dask: Parallel Computation with Blocked algorithms and Task Scheduling”. In: *Proceedings of the 14th Python in Science Conference* (2015), pp. 126–132.

[69] A. B. Yoo, M. A. Jette, and M. Grondona. “SLURM: Simple Linux Utility for Resource Management”. In: *Job Scheduling Strategies for Parallel Processing*. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2003, pp. 44–60. ISBN: 978-3-540-39727-4.

[70] *Slurm Workload Manager - sbatch*. URL: `https://slurm.schedmd.com/sbatch.html` (visited on 01/23/2024).

[71] A. Tiwari and B. Ensing. “Reactive trajectories of the Ru2+/3+ self-exchange reaction and the connection to Marcus’ theory”. In: *Faraday Discussions* 195.0 (Jan. 2017), pp. 291–310. ISSN: 1364-5498.

[72] M. Moqadam et al. “Local initiation conditions for water autoionization”. In: *Proceedings of the National Academy of Sciences* 115.20 (May 2018), E4569–E4576.

[73] K. Lindorff-Larsen et al. “How Fast-Folding Proteins Fold”. In: *Science* 334.6055 (Oct. 2011), pp. 517–520.

[74] O. Aarøen et al. “Thin film breakage in oil–in–water emulsions, a multidisciplinary study”. In: *Colloids and Surfaces A: Physicochemical and Engineering Aspects* 632 (Jan. 2022), p. 127808. ISSN: 0927-7757.

[75] T. V. Sawant et al. “Harnessing Interfacial Electron Transfer in Redox Flow Batteries”. In: *Joule* 5.2 (Feb. 2021), pp. 360–378. ISSN: 2542-4351.

[76] V. Coropceanu et al. “Charge-transfer electronic states in organic solar cells”. In: *Nature Reviews Materials* 4.11 (Nov. 2019), pp. 689–707. ISSN: 2058-8437.

[77] J. Blumberger. “Recent Advances in the Theory and Molecular Simulation of Biological Electron Transfer Reactions”. In: *Chemical Reviews* 115.20 (Oct. 2015), pp. 11191–11238. ISSN: 0009-2665.

[78] R. A. Marcus. “Electrostatic Free Energy and Other Properties of States Having Nonequilibrium Polarization. I”. In: *The Journal of Chemical Physics* 24.5 (May 1956), pp. 979–989. ISSN: 0021-9606.

[79] R. A. Marcus. “On the Theory of Oxidation-Reduction Reactions Involving Electron Transfer. I”. In: *The Journal of Chemical Physics* 24.5 (May 1956), pp. 966–978. ISSN: 0021-9606.

[80] G. King and A. Warshel. “Investigation of the free energy functions for electron transfer reactions”. In: *The Journal of Chemical Physics* 93.12 (Dec. 1990), pp. 8682–8692. ISSN: 0021-9606.

[81] A. D. Clegg et al. “Experimental Validation of Marcus Theory for Outer-Sphere Heterogeneous Electron-Transfer Reactions: The Oxidation of Substituted 1,4-Phenylenediamines”. In: *ChemPhysChem* 5.8 (2004), pp. 1234–1240. ISSN: 1439-7641.

[82] R. A. Kuharski et al. “Molecular model for aqueous ferrous–ferric electron transfer”. In: *The Journal of Chemical Physics* 89.5 (Sept. 1988), pp. 3248–3257. ISSN: 0021-9606.

[83] J. K. Hwang and A. Warshel. “Microscopic examination of free-energy relationships for electron transfer in polar solvents”. In: *Journal of the American Chemical Society* 109.3 (Feb. 1987), pp. 715–720. ISSN: 0002-7863.

[84] N. Marzari et al. "Maximally localized Wannier functions: Theory and applications". In: *Reviews of Modern Physics* 84.4 (Oct. 2012), pp. 1419–1475.

[85] T. D. Kühne et al. "CP2K: An electronic structure and molecular dynamics software package - Quickstep: Efficient and accurate electronic structure calculations". In: *The Journal of Chemical Physics* 152.19 (May 2020), p. 194103. ISSN: 0021-9606.

[86] D. B. Zederkof et al. "Resolving Femtosecond Solvent Reorganization Dynamics in an Iron Complex by Nonadiabatic Dynamics Simulations". In: *Journal of the American Chemical Society* 144.28 (July 2022), pp. 12861–12873. ISSN: 0002-7863.

[87] W. Humphrey, A. Dalke, and K. Schulten. "VMD: Visual molecular dynamics". In: *Journal of Molecular Graphics* 14.1 (Feb. 1996), pp. 33–38. ISSN: 0263-7855.

[88] S. Roet, C. D. Daub, and E. Riccardi. "Chemistrees: Data-Driven Identification of Reaction Pathways via Machine Learning". In: *Journal of Chemical Theory and Computation* 17.10 (Oct. 2021), pp. 6193–6202. ISSN: 1549-9626.

[89] T. S. van Erp et al. "Analyzing Complex Reaction Mechanisms Using Path Sampling". In: *Journal of Chemical Theory and Computation* 12.11 (Nov. 2016), pp. 5398–5410. ISSN: 1549-9626.

[90] B. Dziejarski et al. "CO2 capture materials: a review of current trends and future challenges". In: *Materials Today Sustainability* 24 (Dec. 2023), p. 100483. ISSN: 2589-2347.

[91] J. Y. S. Leung, S. Zhang, and S. D. Connell. "Is Ocean Acidification Really a Threat to Marine Calcifiers? A Systematic Review and Meta-Analysis of 980+ Studies Spanning Two Decades". In: *Small* 18.35 (2022), p. 2107407. ISSN: 1613-6829.

[92] C. Ma, F. Pietrucci, and W. Andreoni. "CO2 Capture and Release in Amine Solutions: To What Extent Can Molecular Simulations Help Understand the Trends?" In: *Molecules* 28.18 (Jan. 2023), p. 6447. ISSN: 1420-3049.

[93] S. Kim et al. "Asymmetric chloride-mediated electrochemical process for CO2 removal from oceanwater". In: *Energy & Environmental Science* 16.5 (May 2023), pp. 2030–2044. ISSN: 1754-5706.

[94] B. M. Szyja, J. Zasada, and E. Dziadyk-Stopyra. "Ru-pincer complexes as charge transfer mediators in electrocatalytic CO2 reduction". In: *Molecular Catalysis* 555 (Feb. 2024), p. 113875. ISSN: 2468-8231.

[95] O. Cheong et al. "Stay Hydrated! Impact of Solvation Phenomena on the CO2 Reduction Reaction at Pb(100) and Ag(100) surfaces". In: *ChemSusChem* 16.21 (2023), e202300885. ISSN: 1864-564X.

[96] E. Boutin et al. "Aqueous Electrochemical Reduction of Carbon Dioxide and Carbon Monoxide into Methanol with Cobalt Phthalocyanine". In: *Angewandte Chemie International Edition* 58.45 (2019), pp. 16172–16176. ISSN: 1521-3773.

[97] S. Lin et al. “Covalent organic frameworks comprising cobalt porphyrins for catalytic CO2 reduction in water”. In: *Science* 349.6253 (Sept. 2015), pp. 1208–1213.

[98] H. Xu et al. “Highly selective electrocatalytic CO2 reduction to ethanol by metallic clusters dynamically formed from atomically dispersed copper”. In: *Nature Energy* 5.8 (Aug. 2020), pp. 623–632. ISSN: 2058-7546.

[99] L. Ruiz Pestana et al. “Ab initio molecular dynamics simulations of liquid water using high quality meta-GGA functionals”. In: *Chemical Science* 8.5 (May 2017), pp. 3554–3565. ISSN: 2041-6520.

[100] S. R. Emerson and J. I. Hedges. *Chemical Oceanography and the Marine Carbon Cycle*. 1st edition. Cambridge: Cambridge University Press, June 2008. ISBN: 978-0-521-83313-4.

[101] *Carbonic_ acid*. URL: `https://www.chemeurope.com/en/encyclopedia/Carbonic_acid.html` (visited on 03/14/2024).

[102] M. T. Nguyen et al. “How Many Water Molecules Are Actively Involved in the Neutral Hydration of Carbon Dioxide?” In: *The Journal of Physical Chemistry A* 101.40 (Oct. 1997), pp. 7379–7388. ISSN: 1089-5639.

[103] T. Mori et al. “Spectroscopic detection of isolated carbonic acid”. In: *The Journal of Chemical Physics* 130.20 (May 2009), p. 204308. ISSN: 0021-9606.

[104] C. E. Tzeliou, M. A. Mermigki, and D. Tzeli. “Review on the QM/MM Methodologies and Their Application to Metalloproteins”. In: *Molecules* 27.9 (Apr. 2022), p. 2660. ISSN: 1420-3049.

[105] D. W. Borhani and D. E. Shaw. “The future of molecular dynamics simulations in drug discovery”. In: *Journal of Computer-Aided Molecular Design* 26.1 (Jan. 2012), pp. 15–26. ISSN: 1573-4951.

[106] L. Grajciar et al. “Towards operando computational modeling in heterogeneous catalysis”. In: *Chemical Society Reviews* 47.22 (Nov. 2018), pp. 8307–8348. ISSN: 1460-4744.

[107] D. Satoh et al. “Folding free-energy landscape of a 10-residue mini-protein, chignolin”. In: *FEBS Letters* 580.14 (June 2006), pp. 3422–3426. ISSN: 0014-5793.

PAPER A

THE JOURNAL OF PHYSICAL CHEMISTRY A
A JOURNAL OF THE AMERICAN CHEMICAL SOCIETY

Exchanging Replicas with Unequal Cost, Infinitely and Permanently

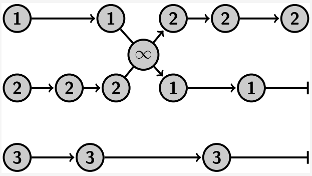
Sander Roet, Daniel T. Zhang, and Titus S. van Erp*

Cite This: *J. Phys. Chem. A* 2022, 126, 8878−8886

Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: We developed a replica exchange method that is effectively parallelizable even if the computational cost of the Monte Carlo moves in the parallel replicas are considerably different, for instance, because the replicas run on different types of processor units or because of the algorithmic complexity. To prove detailed-balance, we make a paradigm shift from the common conceptual viewpoint in which the set of parallel replicas represents a high-dimensional superstate, to an ensemble-based criterion in which the other ensembles represent an environment that might or might not participate in the Monte Carlo move. In addition, based on a recent algorithm for computing permanents, we effectively increase the exchange rate to infinite without the steep factorial scaling as a function of the number of replicas. We illustrate the effectiveness of this replica exchange methodology by combining it with a quantitative path sampling method, replica exchange transition interface sampling (RETIS), in which the costs for a Monte Carlo move can vary enormously as paths in a RETIS algorithm do not have the same length and the average path lengths tend to vary considerably for the different path ensembles that run in parallel. This combination, coined ∞RETIS, was tested on three model systems.



1. INTRODUCTION

The Markov chain Monte Carlo (MC) method is one of the most important numerical techniques for computing averages in high-dimensional spaces, like the configuration space of a many-particle system. The approach has applications in a wide variety of fields ranging from computational physics, theoretical chemistry, economics, and genetics. The MC algorithm effectively generates a selective random walk through state space in which the artificial steps are designed to ensure that the frequency of visiting any particular state is proportional to the equilibrium probability of that state. The Metropolis[1] or the more general Metropolis−Hastings[2] algorithms are the most common approaches for designing such random steps (MC moves) based on the detailed-balance principle. That is, the MC moves should be constructed such that the number of transitions from an old state s^o to a new state s^n is exactly balanced by the number of transitions from the new to the old state: $\rho(s^{(o)})\pi(s^{(o)} \to s^{(n)}) = \rho(s^{(n)})\ \pi(s^{(n)} \to s^{(o)})$, where $\rho(\cdot)$ is the state space equilibrium probability density and $\pi(\cdot)$ are the probabilities to make a transition between the two states given the set of possible MC moves. Further, the transition is split into a generation and an acceptance/rejection step such that $\pi(s \to s') = P_{\text{gen}}(s \to s')\ P_{\text{acc}}(s \to s')$. In the case that the sampled state space is the configuration space of a molecular system at constant temperature, P_{gen} might relate to moving a randomly picked particle in a random direction over a small random distance, and $\rho(s)$ is proportional to the Boltzmann weight $e^{-\beta E(s)}$, with $\beta = 1/k_B T$ the inverse temperature, k_B the Boltzmann constant, and $E(s)$ the state's energy.

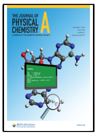
The Metropolis−Hastings algorithm takes a specific solution for the acceptance probability

$$P_{\text{acc}}(s^{(o)} \to s^{(n)}) = \min\left[1, \frac{\rho(s^{(n)}) P_{\text{gen}}(s^{(n)} \to s^{(o)})}{\rho(s^{(o)}) P_{\text{gen}}(s^{(o)} \to s^{(n)})}\right] \quad (1)$$

The generation probabilities will cancel in the above expression if they are symmetric, $P_{\text{gen}}(s \to s') = P_{\text{gen}}(s' \to s)$ as in the less generic Metropolis scheme. At each MC step, the new state is either accepted or rejected based on the probability above. In case of a rejection, the old state is maintained and resampled. This scheme obeys detailed-balance. In addition, if the set of MC moves are ergodic, equilibrium sampling is guaranteed. When ergodic sampling, even if mathematically obeyed, is slowed down by a rough (free) energy landscape, replica exchange MC becomes useful.

Replica exchange MC (or replica exchange molecular dynamics) is based on the idea to simulate several copies of the system with different ensemble definitions,[3−5] most commonly ensembles with increasing temperature (parallel tempering). By performing "swaps" between adjacent replicas, the low-temperature replicas gain access to the broader space region that are explored by the high-temperature replicas. The detailed-balance and corresponding acceptance−rejection step can be derived by viewing the set of states in the different ensembles (replicas) as a single high-dimensional superstate $S = (s_1, s_2, \cdots, s_N)$ representing the system in a set of N

Received: September 2, 2022
Revised: October 18, 2022
Published: November 17, 2022



independent "parallel universes". The Metropolis scheme applied to the superstate yields

$$P_{\mathrm{acc}}(S^{(o)} \rightarrow S^{(n)}) = \min\left[1, \frac{\rho(S^{(n)})}{\rho(S^{(o)})}\right] \tag{2}$$

in which the probability of the superstate equals

$$\rho(S) = \rho(s_1, s_2, \cdots, s_N) = \prod_{i=1}^{N} \rho_i(s_i) \tag{3}$$

where $\rho_i(\cdot)$ is the specific probability density of ensemble i. For example, the move that attempts to swap the first two states, implying $S^o = (s_1, s_2, \cdots, s_N)$ and $S^n = (s_2, s_1, \cdots, s_N)$, will be accepted with a probability

$$P_{\mathrm{acc}} = \min\left[1, \frac{\rho_1(s_2)\rho_2(s_1)}{\rho_1(s_1)\rho_2(s_2)}\right] \tag{4}$$

In a replica exchange simulation, swapping moves and standard MC or MD steps are applied alternately. Parallel computing will typically distribute the same number of processing units per ensemble to carry out the computationally intensive standard moves. The swapping move is cheap but requires that the ensembles involved in the swap have completed their previous move. If the standard moves in each ensemble require different computing times, then several processing units have to wait for the slow ones to finish.

If the disbalance per move is relatively constant, then the replicas could effectively be made to progress in cohort by trying to differentiate the number of processing units per ensemble or the relative frequency of doing replica exchange versus standard moves per ensemble. However, this disbalance is not constant in several MC methods, such as with configurational bias MC[6−8] or path sampling.[9] The number of elementary steps to grow a polymer in configurational bias MC obviously depends on the polymer's length that is being grown, but also early rejections lead to a broad distribution of the time it takes to complete a single MC move even in uniform polymer systems. Analogously, the time required to complete an MC move in path sampling simulations will depend on the length of the path being created. Other examples of complex Monte Carlo methods with a fluctuating CPU cost per move are cluster Monte Carlo algorithms[10] and event-chain Monte Carlo.[11,12]

We will show that the standard acceptance eqs 1 and 4 can be applied in a parallel scheme in which ensembles are updated irregularly in time and the average frequency of MC moves is different for the ensembles. In addition, we show that we can apply an infinite swapping[13] scheme between the available ensembles. For this, we develop a new protocol based on the evaluation of permanents that circumvents the steep factorial scaling. This last development is also useful for standard replica exchange.

2. METHODS

2.1. Finite Swapping. In the following, we will assume that we have two types of MC moves. One move that is CPU-intensive and can be carried out within a single ensemble, and replica exchange moves between ensembles which are relatively cheap to execute. The CPU-intensive move will be carried out by a single worker (one processor unit, one node, or a group of nodes) and these workers perform their task in parallel on the different ensembles. One essential part of our algorithm is that we have less workers than ensembles such that whenever the worker is finished and produced a new state for one ensemble, this state can directly be swapped with the states of any of the available ensembles (the ones not occupied by a worker). After that, the worker will randomly switch to another unoccupied ensemble for performing a CPU-intensive move.

In its most basic form, the algorithm consists of the following steps:

1. Define N ensembles and let $\rho_i(\cdot)$ be the probability distribution of ensemble i. We also define P_{RE} which is the probability of doing a replica exchange move.
2. Assign $K < N$ workers (processing units) to K of the N ensembles for performing a CPU-intensive MC move. Each ensemble is at all times occupied by either 1 or 0 workers. The following steps are identical for all of the workers.
3. If the worker is finished with its MC move in ensemble i, the new state is accepted or rejected according to eq 1 (with ρ_i for ρ). Ensemble i is updated with the new state (or by resampling the old state in case of rejection) and is then considered to be free.
4. Take a uniform random number ν between 0 and 1. If $\nu > P_{\mathrm{RE}}$, go to step 7.
5. Among the free ensembles, pick a random pair (i, j).
6. Try to swap the states of ensembles i and j using eq 4 (with labels i, j instead of 1, 2). Update ensembles i, j with the swapped state or the old state in case of a rejection. Return to step 4.
7. Select one of the free ensembles at random and assign the worker to that ensemble for performing a new standard move. Go to step 3.

In this algorithm, ensembles are not updated in cohort like in standard replica exchange, but updates occur at irregular intervals. In addition, the different ensemble conditions can result in systematic differences in the number of states that are being created over time. To prove that the above scheme actually samples the correct distributions requires a fundamentally new conceptual view as the superstate picture is no longer applicable. Despite that the algorithm uses the same type of eqs 1 and 4, as one would use in standard replica exchange, it does not rely on eqs 2 and 3 that are no longer valid. In the SI, we provide a proof from the individual ensemble's perspective in which the other ensembles provide an "environment" $\mathcal{E}$ that might, or might not, participate in the move of the ensemble considered. By doing so, we no longer require that the number of transitions from old to new, $S^{(o)} \rightarrow S^{(n)}$, is the same as from new to old, $S^{(n)} \rightarrow S^{(o)}$. Instead, by writing $S = (s_1, \mathcal{E})$, from ensemble 1's perspective, we have that the number of $(s_1^{(o)}, \mathcal{E}^{(o)}) \rightarrow (s_1^{(n)}, {}^{a}\mathcal{E}^{(n)})$ transitions should be equal to the number of $(s_1^{(n)}, \mathcal{E}^{(o)}) \rightarrow (s_1^{(o)}, {}^{a}\mathcal{E}^{(n)})$ transitions when the standard move is applied where ${}^{a}\mathcal{E}^{(n)}$ refers to *any* new environment. In the SI, we show a similar detailed-balance condition for the replica exchange moves. At step 6 we sample only ensemble i and j or, alternatively, all free ensembles get a sample update. This would mean resampling the existing state of those not involved in a swap ("null move"). This makes the approach more similar to the superstate sampling albeit using only free ensembles, as described in the SI. The null move does not reduce the statistical uncertainty, but we mention it here as it makes it easier to explain the infinite swapping approach. But for the detailed-balance conditions to be valid it is imperative that occupied ensembles are not sampled.

An essential aspect of the efficiency of our algorithm is that the number of workers K is less than the number of ensembles N. The case $K = N$ is valid but would reduce the number of replica exchange moves to zero as only one ensemble is free at the maximum. Reducing the K/N ratio will generally imply a higher acceptance in the replica exchange moves as we can expect a higher number of free ensembles whose distributions have significant overlap. What gives the optimum number of workers is therefore a nontrivial question that we will further explore in Section 4. However, for case K < N we can maximize the effect of the replica exchange moves by taking the P_{RE} parameter as high as possible. In fact, we can simulate the effect of the limit $P_{\mathrm{RE}} \to 1$ without having to do an infinite number of replica exchange moves explicitly. This leads to an infinite swapping[13] version of our algorithm.

2.2. Infinite Swapping. If in the previously described algorithm we take $P_{\mathrm{RE}} = 1 - \delta$, we will loop through steps 4−6 for many iterations ($n_{\mathrm{it}} = \sum_{n=0}^{\infty} n(1-\delta)^n\delta = 1/\delta$ in the limit $\delta \to 0$) before getting to step 7. When δ vanishes and n_{it} becomes infinitely large, we expect that all possible swaps will be executed an infinite number of times. Since the swaps obey detailed-balance between unoccupied ensembles, these will essentially sample the distribution of eq 3 (for the subset S^* of unoccupied ensembles). Hence, when the loop is exited, each possible permutation $\sigma \in S^*$ has been sampled $n_{\mathrm{it}} \times \rho(\sigma)/\sum_\sigma \rho(\sigma)$ times. By lumping all the times that the same permutation was sampled and normalizing by division with n_{it}, we simply sample all of the possible permutations in one go using fractional weights that sum up to 1. This is then the only sampling step, as the single update in step 3 can be skipped due to its negligible $1/n_{\mathrm{it}}$ weight.

The idea of doing an "infinite number" of swapping moves has been proposed before,[13−16] but here we give a different flavor to this approach by a convenient reformulation of the problem into permanents that allows us to beat the steep factorial scaling reported in earlier works.[13] The permanent formulation goes as follows. Suppose that after step 3, there are four free ensembles (we name them e_1, e_2, e_3, e_4) containing four states (s_1, s_2, s_3, s_4). Which state is in which ensemble after this step is irrelevant. We can now define a weight-matrix W

$$W = \begin{array}{c|cccc} & e_1 & e_2 & e_3 & e_4 \\ \hline s_1 & W_{11} & W_{12} & W_{13} & W_{14} \\ s_2 & W_{21} & W_{22} & W_{23} & W_{24} \\ s_3 & W_{31} & W_{32} & W_{33} & W_{34} \\ s_4 & W_{41} & W_{42} & W_{43} & W_{44} \end{array}$$

where $W_{ij} \propto \rho_j(s_i)$. Essential to our approach is the computation of the permanent of the W-matrix, perm(W), and that of the $W\{ij\}$-matrices in which the row i and column j are removed.

The permanent of a matrix is similar to the determinant but without alternating signs. We can, henceforth, write perm(W) $= \sum_{j=1}^{4} W_{1j}\mathrm{perm}(W\{1j\})$. As the permanent of the 1 × 1 matrix is obviously equal to the single matrix value, the permanent of arbitrary dimension could in principle be solved recursively using this relation. Based on the permanents of W, we will construct a probability matrix P

$$P = \begin{array}{c|cccc} & e_1 & e_2 & e_3 & e_4 \\ \hline s_1 & P_{11} & P_{12} & P_{13} & P_{14} \\ s_2 & P_{21} & P_{22} & P_{23} & P_{24} \\ s_3 & P_{31} & P_{32} & P_{33} & P_{34} \\ s_4 & P_{41} & P_{42} & P_{43} & P_{44} \end{array}$$

where P_{ij} is the chance to find state s_i in ensemble e_j. As for each permutation each state is in one ensemble and each ensemble contains one state, the P-matrix is bistochastic: both the columns and the rows sum up to 1. If we consider S^*_{ij} the set of permutations in which state s_i is in e_j, we can write $P_{ij} = \sum_{\sigma \in S^*_{ij}} \rho(\sigma)/\sum_{\sigma' \in S}{}^*\rho(\sigma')$. We can, however, also use the permanent representation in which

$$P_{ij} = \frac{W_{ij}\mathrm{perm}(W\{ij\})}{\mathrm{perm}(W)} \tag{5}$$

So far we have not won anything as computing the permanent via the recursive relation mentioned above has still the factorial scaling. The Gaussian elimination approach, which allows an order $O(n^3)$ computation for determinants of $n \times n$ matrices, will not work for permanents as only some but not all row and column operations have the same effect to a permanent as to a determinant. One can for instance swap rows and columns without changing the permanent. Multiplying a row by a nonzero scalar multiplies the permanent by the same scalar. Hence, this will not affect the P-matrix based on eq 5. Unlike the determinant, adding or subtracting to a row a scalar multiple of another row, an essential part of the Gaussian elimination method, does change the permanent. This makes the permanent computation of a large matrix excessively more expensive than the computation of a determinant. Yet, recent algorithms based on the Balasubramanian-Bax-Franklin-Glynn (BBFG) formula[17−20] scale as $O(2^n)$. This means that the computation of the full P-matrix scales as $O(2^n \times n^2)$, which seems still steep but is nevertheless a dramatic improvement compared to factorial scaling.

For our target time of 1 second, for instance, we could only run the algorithm up to $N = 7$ in the factorial approach, while we reach $N = 12$ in the BBFG method using a mid-to-high-end laptop (DELL XPS 15 with an Intel Core i7-8750H). If matrix size of $N = 20$ is the target, the BBFG method can perform a full P-matrix determination in ~711 s, while it would take ~15.3 × 10^6 years in the factorial approach. The BBFG method is the fastest completely general solution for the problem of computing a P-matrix from any W-matrix. For several algorithms, the W-matrix has special characteristics that can be exploited to further increase efficiency. For instance, if by shuffling the rows and columns the W-matrix can be made into a block form, where squared blocks at the diagonal have only zeros at their right and upper side, the permanent is equal to the product of the block's permanents. For instance, if $W_{14} = W_{24} = W_{34} = 0$, we have two blocks, 3 × 3 and 1 × 1. If $W_{13} = W_{14} = W_{23} = W_{24} = 0$, we can identify two blocks of 2 × 2, etc. Identification of blocks can hugely decrease the computation of a large permanent. Another speed-up can be made if all rows in the W-matrix are a sequence of ones followed by all zeros, or can be made into that form after the previously mentioned column and row operations. This makes an order $O(n^2)$ approach possible. We will further discuss this in Section 3.1.

The infinite swapping approach changes the aforementioned algorithm from step 3:

3 If the worker is finished with its MC move in a specific ensemble, the new state is accepted or rejected (but not yet sampled) according to eq 1. The ensemble is free.

4 Determine the W-matrix based on all unoccupied ensembles, calculate the P-matrix based on eq 5, and update all of the unoccupied ensembles by sampling all free states with the fractional probabilities corresponding to the columns in the P-matrix.

5 Pick randomly one of the free ensembles e_j.

6 Pick one of the available states (s_1, s_2, ⋯) based on a weighted random selection in which state s_i has a probability of P_{ij} to be selected.

7 The worker is assigned to do a new standard move in ensemble e_j based on previous state s_i. Go to step 3.

3. APPLICATION: ∞RETIS

Replica Exchange Transition Interface Sampling (RETIS)[21,22] is a quantitative path sampling algorithm in which the sampled states are short molecular trajectories (paths) with certain start and end conditions, and a minimal progress condition. New paths are being generated by a Monte Carlo move in path space, such as the shooting move[23] in which a randomly selected phase point of the previous path is randomly modified and then integrated backward and forward in time by means of molecular dynamics (MD). The required minimal progress increases with the rank of the ensemble such that the final ensemble contains a reasonable fraction of transition trajectories. The start and end conditions, as well as the minimal progress, are administered by the crossings of interfaces (λ_0, λ_1, ⋯, λ_M) with $\lambda_{k+1} > \lambda_k$, that can be viewed as nonintersecting hypersurfaces in phase space having a fixed value of the reaction coordinate. A MC move that generates a trial path not fulfilling the path ensemble's criteria is always rejected. RETIS defines different path ensembles based on the direction of the paths and the interface that has to be crossed, but all paths start by crossing λ_0 (near the reactant state/state A) and they end by either crossing λ_0 again or reaching the last interface λ_M (near the product state/state B). There is one special path ensemble, called $[0^-]$, that explores the left side of λ_0, the reactant well, while all other path ensembles, called $[k^+]$ with $k = 0, 1, \cdots M - 1$, start by moving to the right from λ_0 reaching at least λ_k.

A central concept in RETIS is the so-called overall crossing probability, the chance that a path that crosses λ_0 in the positive direction reaches λ_M without recrossing λ_0. It provides the rate of the process when multiplied with the flux through λ_0 (obtained from the path lengths in $[0^-]$ and $[0^+]$[22]) and is usually an extremely small number. The chance that any of the sampled paths in the $[0^+]$ path ensemble crosses λ_M is generally negligible, but a decent fraction of those (∼0.1−0.5) will cross λ_1 and some even λ_2. Likewise, paths in the $[k^+]$, $k > 0$, path ensembles have a much higher chance to cross λ_{k+1} than a $[0^+]$-path as they already cross λ_k. This leads to the calculation of M local conditional crossing probabilities, the chance to cross λ_{k+1} given λ_k was crossed for $k = 0,1, \cdots, M - 1$, whose product gives an exact expression for the overall crossing probability with an exponentially reduced CPU cost compared to MD.

The efficiency is further hugely improved by executing replica exchange moves between the path ensembles. These swaps are essentially cost-free since there is no need to simulate additional ensembles that are not already required. An accepted swapping move in RETIS provides new paths in two ensembles without the expense of having to do MD steps. The enhancement in efficiency is generally even larger than one would expect based on these arguments alone as path ensembles higher-up the barrier provide a similar effect as the high-temperature ensembles in parallel tempering. In addition, point exchange moves between the $[0^-]$ and $[0^+]$ ensembles are performed by exchanging the end and start points of these paths that are then continued by MD at the opposite site of the λ_0 interface.

While TIS[24] (without replica exchange) can run all path ensembles embarrassingly parallel, the RETIS algorithm increases the CPU-time efficiency but is difficult to parallelize and open source path sampling codes, like OpenPathSampling[25] and PyRETIS,[26] implement RETIS as a fully sequential algorithm. The path length distributions are generally broad with an increasing average path length as a function of the ensemble's rank. This becomes increasingly problematic the more ensembles you have as they all have to wait for the slowest ensemble. This means that while RETIS will give the best statistics per CPU-hour, it might not give the best statistics in wall-time. Our parallel scheme can effectively deal with the unequal CPU cost of the replicas, which allows us to increase the wall-time efficiency with no or minimal reduction in CPU-time efficiency. On the contrary, our method does not give an equal distribution of the CPU-time over the different ensembles nor an equal number of samples per ensemble, leading to a smarter distribution of CPU hours. The CPU efficiency therefore even seems to improve slightly.

3.1. *W*-Matrix in RETIS.

If there are $M + 1$ interfaces, λ_0, λ_1, ⋯, λ_M, there are also $N = M + 1$ ensembles, $[0^-]$, $[0^+]$, $[1^+]$, ⋯, $[(M - 1)^+]$. For K workers, the size of the W-matrix is, hence, either $(N - K + 1) \times (N - K + 1)$ or $(N - K) \times (N - K)$ as swappings are executed when 1 of the K workers is free, while the remaining $K - 1$ workers occupy path ensembles that are locked and do not participate in the swap. The smallest matrix occurs when one worker is occupying both $[0^-]$ and $[0^+]$ during the point exchange move, as described in the SI.

Paths can be represented by a sequence of time slices, the phase points visited by the MD trajectory. For a path of length $L + 1$, $X = (x_0, x_1, \cdots, x_L)$, the plain path probability density $\rho(X)$ is given by the probability of the initial phase point times the dynamical transition probabilities to go from one phase point to the next: $\rho(X) = \rho(x_0)\phi(x_0 \rightarrow x_1)\phi(x_1 \rightarrow x_2) \cdots \phi(x_{L-1} \rightarrow x_L)$. Here, the transition probabilities depend on the type of dynamics (deterministic, Langevin, Nosé-Hoover dynamics, etc). The weight of a path within a specific path ensemble $\rho_j(X)$ can be expressed as the plain path density times the indicator function $\mathbf{1}_{e_j}$ and possibly an additional weight function $w_j(X)$: $\rho_j(X) = \rho(X) \times \mathbf{1}_{e_j}(X) \times w_j(X)$. The indicator function equals 1 if path X belongs to ensemble e_j. Otherwise, it is 0. The additional weight function $w_j(X)$ is part of the high-acceptance protocol that is used in combination with the more recent path generation MC moves such as stone skipping[27] and wire fencing.[28] Using these "high-acceptance weights", nearly all of the CPU-intensive moves can be accepted as they are tuned to cancel the P_{gen}-terms in the Metropolis−Hastings scheme, eq 1, and the effect of the

Table 1. Results of the Three Model Systems Showing Crossing Probabilities (P_{cross}), Permeabilities (perm.), and Rates for Different Number of Workers (K)[a]

MSVS		two-channel system			double well with wire fencing		
K	$P_{\text{cross}}/10^{-50}$	K	$P_{\text{cross}}/\ 10^{-5}$	perm./10^{-6}	K	$P_{\text{cross}}/10^{-7}$	rate/10^{-7}
1	0.61 ± 0.33	1	1.52 ± 0.17	1.28 ± 0.14	1	5.91 ± 0.18	2.59 ± 0.07
5	1.47 ± 1.04	2	1.63 ± 0.24	1.37 ± 0.20	2	5.70 ± 0.13	2.51 ± 0.06
10	0.86 ± 0.51	3	1.52 ± 0.07	1.28 ± 0.06	3	5.57 ± 0.19	2.45 ± 0.08
15	0.68 ± 0.08	4	1.42 ± 0.10	1.19 ± 0.08	4	5.20 ± 0.30	2.34 ± 0.12
20	1.02 ± 0.13	5	1.40 ± 0.12	1.18 ± 0.10	5	5.05 ± 0.41	2.23 ± 0.18
25	1.02 ± 0.17	6	1.54 ± 0.06	1.30 ± 0.05	6	5.49 ± 0.29	2.42 ± 0.13
30	1.26 ± 0.24	7	1.48 ± 0.08	1.24 ± 0.07	7	4.99 ± 0.39	2.21 ± 0.17
35	1.05 ± 0.15	8	1.46 ± 0.08	1.23 ± 0.06	8	4.88 ± 0.43	2.15 ± 0.19
40	1.05 ± 0.14	9	1.42 ± 0.10	1.20 ± 0.08			
45	0.93 ± 0.09	10	1.44 ± 0.08	1.21 ± 0.07			
50	1.00 ± 0.07	11	1.41 ± 0.09	1.19 ± 0.08			
		12	1.30 ± 0.15	1.09 ± 0.12			
			Literature/Theoretical Result				
			1.23 ± 0.16[b]	1.06 ± 0.14[b]			2.79 ± 0.70[d]
						5.84 ± 0.13[e]	2.58 ± 0.06[e]
	1.00[a]		1.61[c]	1.37[c]		5.83[c]	2.58[c]

[a] All results are shown in dimensionless units. Errors are based on single standard deviations. Values shown in the lower part are a: exact result, b: ref 31, c: approximated value based on Kramers' theory (see the SI), d: ref 32, and e: ref 28.

nonphysical weights is undone in the analysis by weighting each sampled path with the inverse of $w_j(X)$.

While the path probability $\rho(s_i = X)$ is difficult to compute, determining $\mathbf{1}_j(s_i)$ and $w_j(s_i)$ is trivial. It is therefore a fortunate coincidence that we can replace $W_{ij} = \rho_j(s_i)$ with

$$W_{ij} = \mathbf{1}_{e_j}(s_i) w_j(s_i) \tag{6}$$

because the P-matrix does not change if we divide or multiply a row by the same number, as mentioned in Section 2. Except for $[0^-]$, all path ensembles have the same start and end conditions and only differ with respect to the interface crossing condition. A path that crosses interface λ_k automatically crosses all lower interfaces $\lambda_{l<k}$. Reversely, if the path does not cross λ_k, it will not cross any of the higher interfaces $\lambda_{l>k}$. This implies that if the columns of W_{ij} are ordered such that the first column (e_1) is the first available ensemble from the sequence ($[0^-]$, $[0^+]$, $[1^+]$, $\cdots$, $[(M-1)^+]$), the second column (e_2) is the second available ensemble, and so on, most rows will end with a series of zeros.

Reordering the rows with respect to the number of trailing zeros, almost always ensures that the W-matrix can be brought into a block form such that the permanent can be computed faster based on smaller matrices. In particular, if $[0^-]$ is part of the free ensembles, it will always form a 1×1 block as there is always one and no more than one available path that fits in this ensemble.

If high acceptance is not applied, we have $w_j(X) = 1$ and each row in the W-matrix (after separating the $[0^-]$ ensemble if it is part of the free ensembles) is a sequence of ones followed by all zeros. The W-matrix can hence be represented by an array $(n_1, n_2, n_3, \cdots n_n)$, where each integer n_i indicates the number of ones in row i. As we show in the SI, the permanent of such a W-matrix is simply the product of $(n_i + 1 - i)$: $\text{perm}(W) = \prod_i (n_i + 1 - i)$. Further, the P-matrix can be constructed from the following order $O(n^2)$ method.

The first step is to order the rows of the W-matrix such that $n_1 \le n_2 \le \cdots \le n_n$. We then fill in the P-matrix from top to bottom for each row using

$$P_{ij} = \begin{cases} 0, \text{ if } W_{ij} = 0 \\ \dfrac{1}{n_i + 1 - i}, \text{ if } W_{ij} = 1 \text{ and } [W_{(i-1)j} = 0 \text{ or } i = 1] \\ \left(\dfrac{n_{i-1} + 1 - i}{n_i + 1 - i}\right) P_{(i-1)j}, \text{ otherwise} \end{cases} \tag{7}$$

The approach is extremely fast and allows the computation of P-matrices from a large W-matrix, up to several thousands, within a second of CPU-time. The above method applies whenever the rows of the W-matrix can be transformed into sequence of ones followed by all zeros. Besides RETIS without high acceptance, this would apply to other MC methods like subset sampling[29] or umbrella sampling[30] with semi-infinite rectangular windows.

4. RESULTS AND DISCUSSION

To test our algorithms we ran three types of ∞RETIS simulations. First, a memoryless single variable stochastic (MSVS) process was simulated to mimic a RETIS simulation in which the average path length increases linearly with the rank of the ensemble. A "path" is created by drawing 2 random numbers where the first determines how much progress a path makes and the second determines the path length. These two outcomes are variable and depend on the rank of the ensemble such that the fictitious path in ensemble $[k^+]$ has a 0.1 probability to cross λ_{k+1} and has an average path length of approximately $k/10$ seconds (see Section 2). The worker is paused for a number of seconds equal to the path length before it can participate in replica exchange moves to mimic the time it would take to perform all of the necessary MD steps. While this artificial simulation allows us to investigate the potential strength of the method to tackle extremely rare events, it cannot reveal the effect of correlations between accepted paths when fast exploration of the reaction coordinate's orthogonal directions is crucial. To analyze this effect, we also ran a two-dimensional (2D) membrane permeation system with two

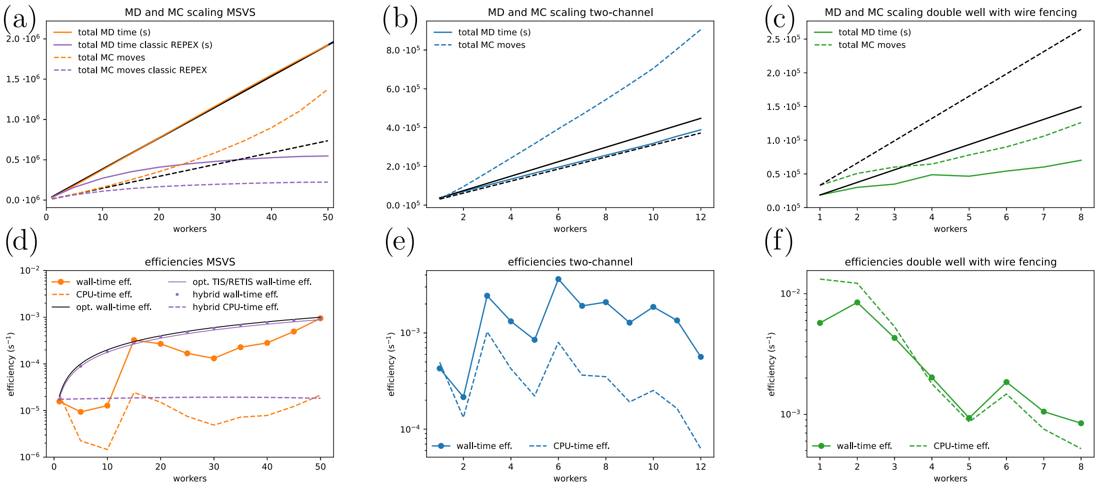


Figure 1. Average scaling of total MD time (cumulative time spent by all of the workers) (solid) and MC moves (dashed) (a−c) and wall-time (solid) and the CPU-time (dashed) efficiencies (d−f) for each number of workers. This is shown for the memoryless single variable stochastic (MSVS) process (a, d, orange), the two-channel system (b, e, blue), and the double well with wire fencing (c, f, green) simulations. Each of the data points is based on five independent simulations. For the scaling plots, the black lines are guides for linear scaling from the 1 worker data-point. The purple lines in the scaling plot for the MSVS simulations (a) show what the scaling would be if we had to wait for the slowest ensemble to finish for each MC move. The black line, purple line, purple dashed line, and points in the efficiency plot of the MSVS process (d) show the optimal, optimal TIS/RETIS, hybrid CPU-time efficiency, and hybrid wall-time efficiency, respectively, as computed in the SI.

slightly asymmetric channels.[31] Finally, to study our algorithm with a more generic W-matrix that needs to be solved via the BBFG formula, we also ran a set of underdamped Langevin simulations of a particle in a double-well potential[32] using the recent wire fencing algorithm with the high-acceptance protocol.[28] All simulation results were performed using five independent runs of 12 h. Errors were based on the standard deviations from these five simulations, except for the MSVS process, where a more reliable statistical error was desired for the comparison with analytical results. Here, block errors were determined on each of the five simulations based on the running average of the overall crossing probability. The block errors were finally combined to obtain the statistical error in the average of the five simulations.

4.1. Memoryless Single Variable Stochastic (MSVS) Process.

Table 1 reports the overall crossing probabilities and their statistical errors for a system with 50 interfaces and 1, 5, 10, 15, 20, 25, 30, 35, 40, 45, and 50 workers. All values are within a 50% deviation from the true value of 10^{-50} with the more accurate estimates for the simulations having a large number of workers. Also, the true value is within one standard deviation of the reported averages for 70% of the data points, as is expected from the standard Gaussian confidence intervals. Figure 1a shows the scaling of the MD time (solid lines) and number of MC moves (dashed lines) of the MSVS simulations (orange) compared to linear scaling (black) and the expected scaling for standard replica exchange (REPEX) in which ensembles are updated in cohort (purple).

Although the number of "MD steps" and MC moves quickly levels off to a nearly flat plateau in the standard approach due to workers being idle as they need to wait for the slowest worker, the replica exchange approach developed in this article shows a perfect linear scaling with respect to the MD time. The number of MC moves in the new method shows an even better than linear scaling due to the fact that the ensembles with shorter "path lengths" get simulated relatively more often with more workers, resulting in more MC moves per second. This in itself does not necessarily mean that the simulations converge much faster because the additional computational effort may not be targeted to the sampling where it is needed. If we neglect the fact that path ensemble simulations are correlated via the replica exchange moves, we can write that the relative error in overall crossing probability ϵ follows from the relative errors in each path ensemble ϵ_i via: $\epsilon^2 = \sum_i \epsilon_i^2$. It is henceforth clear that additional computational power should not aim to lower the error in a few path ensembles that were already low compared to other path ensembles. We therefore measure the effectiveness of the additional workers by calculating computational efficiencies. The efficiency of a specific computational method is here defined as the inverse computer time, CPU- or wall-time, to obtain an overall relative error equal to 1: $\epsilon = 1$.

In Figure 1d, the efficiencies based on wall-time (solid) and CPU-time (dashed) are plotted for the MSVS process. These plots depend on the ability of computing reliable statistical errors in the overall crossing probability that is an extremely small number, 10^{-50}. The somewhat fluctuating behavior of these curves should hence be viewed as statistical noise as the confidence interval of these efficiencies depends on the statistical error of this error. Despite that, clear trends can be observed in which the CPU-time efficiency is more or less constant within statistical fluctuations, while the wall-time efficiency shows an upward trend. If we neglect the effect of replica exchange moves on the efficiency, we can relate these numerical results with theoretical ones[22,33] for any possible division of a fixed total CPU-time over the different ensembles. A common sense approach would be to aim for the same error ϵ_i in each ensemble (which implies doing the same number of MC moves per ensemble) or to divide the total CPU-time evenly over the ensembles. These two strategies correspond to the case $K = 1$ or standard RETIS and $K = N$ or standard TIS, respectively. Ref 33 showed that these two strategies provide the same efficiency, and in the SI, we derive that this leads to a wall-time efficiency as a function of the number of workers (K) equal to $K/56250$, which is denoted by the continuous purple line in Figure 1d (the hypothetical wall-time efficiency for a parallel simulation that uses the CPU hours equally efficient as TIS and serial RETIS, see eq 50 in the SI). The optimum division, however, would give a slightly better wall-time efficiency equal to $K/50{,}000$ which is the continuous black line in this figure. Also shown in Figure 1d are the expected theoretical efficiencies based on the numerical distribution of MC moves in each ensemble. This hybrid numerical/theoretical result is shown by the small purple dots and the dashed purple line. This shows that ∞RETIS, at least for a system in which the path length grows linearly with the ensemble's rank, naturally provides a division of the computational resources that is even better than TIS ($K = N$) or RETIS ($K = 1$). Yet, due to statistical inaccuracies this is only evident for the $K = 15$ case if we base our analysis on the numerical block errors.

In any case, it shows that the possible concern that additional CPU resources in parallel ∞RETIS runs may not be properly targeted due to oversampling of the ensembles with the shorter paths is unfounded. On the contrary, even the CPU efficiency seems to improve slightly compared to TIS and RETIS that have the same CPU efficiency. This is maybe not so surprising since the division of CPU hours over the different ensembles in ∞RETIS is somewhat in between the divisions what one gets with TIS and RETIS, and the optimum[33] is also in between TIS and RETIS. In addition, we leave the number of interfaces and their locations unchanged in our analysis. However, the flexibility of ∞RETIS makes it very easy to add additional interfaces. By placing a higher density of interfaces high on the barrier, it is also possible to target more CPU to the expensive ensembles. This higher density has the additional benefit that the local crossing probabilities in this area are increased so that fewer paths are needed to calculate them.

Coming back to Figure 1d, we see that the best wall-time efficiency is obtained for the case $K = N$, which is essentially equivalent of running independent TIS simulations (i.e., without doing any replica exchange moves). We do not expect this to apply to more complex systems where the replica exchange move is a proven weapon for efficient sampling.

4.2. Two-Channel Simulations. In the middle column of Table 1, we report the calculated crossing probabilities and permeabilities for five simulations for every number of workers. All simulations are somewhat higher, though still in good agreement with the previous simulation from ref 31. We also evaluated the approximate result based on Kramers' theory (see the SI), which seems to confirm the results obtained in this paper. Figure 1b shows the scaling of the MD time (solid lines) and number of MC moves (dashed lines) of the two-channel simulations (blue) compared to linear scaling (black). We see a slightly worse than linear scaling of the MD time, which might just be due to a small positive fluctuation of the 1 worker data-point. We also see a similar more than linear scaling in the number of MC moves as with the MSVS simulations, for the same reason. In Figure 1e, the efficiencies based on wall-time (solid) and CPU-time (dashed) are plotted for the two-channel system. The CPU-time efficiency is more or less flat until 8 workers after which it starts to drop off. The wall-time efficiency shows an upward trend until 10 workers after which it starts to drop off as well. We assign this drop to the reduction of replica exchange moves which is an essential aspect for sampling this system efficiently.[31] This is tangible from Figure S1 in the SI where we plot fraction of trajectories, passing through λ_{M-1}, that are in the lower barrier channel. While from the average fraction it still looks like the simulations sampled both channels for any number of workers, 4 out of the 5 simulations in the $K = N = 12$ case solely visited one of the two channels. This is in agreement with previous TIS results.[31] The $K = 11$ case already provides a dramatic improvement, but is still expected to be suboptimal due to the relatively low frequency of replica exchange moves compared to $K < 11$. From this 2D system, it would indicate that having $K \approx N/2$ is a safe bet for optimum efficiency.

4.3. Double-Well 1D Barrier Using Wire Fencing. In the right column of Table 1, we report the calculated crossing probabilities and rates for the underdamped Langevin particle in the 1D double-well potential. All simulations are in reasonable agreement with each other and the results of refs 28 and 32, as well as the approximate value based on Kramers' theory. However, while these results confirm the soundness of the method, the scaling and efficiency are less convincing. Figure 1c shows a significantly worse than linear scaling. On further inspection, we found the average time per MC move was significantly smaller than our infinite swapping time (1 s) when the simulation was run with more than two workers. This results in a bottleneck on how many MC moves can be started per second, which is the reason for the observed bad scaling. It still is slightly positive instead of flat as the infinite swapping procedure becomes quicker with more workers due to the smaller W-matrix. The same bottleneck can be seen in Figure 1f where both efficiencies plummet with more than two workers. The reported scaling deficiency is of little significance for actual molecular systems where the creation of a full path takes minutes to hours rather than subseconds.

5. CONCLUSIONS

We developed a new generic replica exchange method that is able to effectively deal with MC moves with varying CPU costs, for instance, due to the algorithmic complexity of the MC moves. An essential aspect of the method is that the number of workers, who execute the ensemble's specific MC moves in parallel, is less than the number of ensembles. Once a worker is finished with its move, replica exchange moves are carried out solely between those ensembles that are not occupied by a worker. This implies that the ensembles are updated at irregular intervals and a different number of MC moves will be executed for each ensemble. As a result, the conceptual viewpoint in which the set of replicas are viewed as a single superstate is no longer valid and the existence of some kind of detailed-balance relation is no longer trivial. To prove the exactness of our approach, we introduced new conceptual views on the replica exchange methodology that is different from the common superstate principle. Instead, we show that the distributions in the new approach are conserved for each ensemble individually via a twisted detailed-balance relation in which the other ensembles constitute an environment that is potentially actively involved in the MC move of the considered ensemble. In addition, the method can be combined with an infinite swapping approach without factorial scaling based on a mathematical reformulation using permanents.

We applied the novel replica exchange technique on a path sampling algorithm, RETIS, which is a prototype of algorithm where the costs for a Monte Carlo move can vary enormously. The resulting new path sampling algorithm, coined ∞RETIS, was thereafter tested on three model systems. The results of these simulations show that the number of MD steps increases linearly with the number of workers invoked as long as the ensemble's MC move has a lower computational cost than the replica exchange move carried out by the scheduler. The number of executed MC moves shows an even better than linear scaling. Moreover, the efficiency increases linearly with the number of workers for a low-dimensional system in which the replica exchange move has little effect, while it has an optimum in more complex systems as the number of successful replica exchange moves decreases when the number of workers is close to the number of ensembles.

In summary, the replica exchange method discussed in this paper has a clear potential to accelerate present path sampling simulations, but can also be combined with many other complex algorithms including those that are yet to be invented. With the continuing trend to run progressively more massively parallel computing jobs, our algorithm is likely to gain importance. In the case of ∞RETIS, we envision many applications in the fields of nucleation, self-assembly, chemical reactions, enzymatic catalysis, membrane permeation, protein folding, and other conformational changes in biomolecules. The ∞RETIS method and the noncohort replica exchange method, in general, are therefore expected to open up new avenues in the field of molecular simulations and maybe even beyond.

ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jpca.2c06004.

> Details of the simulations, proof that the replica exchange method with cost unbalanced replicas conserves the equilibrium distribution at the individual ensemble level; $O(n^2)$ algorithm for computing the P-matrix from a W-matrix for the case that the W-matrix consists of rows having a series with ones, followed by zeros; derivations of the theoretical results on the crossing probabilities, rate constant and permeability via Kramers' theory that are shown in Table 1; discussion on the computational efficiencies with the derivations for the most optimal efficiencies; some additional simulation results on the relative transition probabilities through the lower and higher barrier channel for the two-channel system; our current code with which the model system calculations were done and the project started for advanced simulations (PDF)

AUTHOR INFORMATION

Corresponding Author

Titus S. van Erp − *Department of Chemistry, Norwegian University of Science and Technology (NTNU), N-7491 Trondheim, Norway;* orcid.org/0000-0001-6600-6657; Email: titus.van.erp@ntnu.no

Authors

Sander Roet − *Department of Chemistry, Norwegian University of Science and Technology (NTNU), N-7491 Trondheim, Norway;* orcid.org/0000-0003-0732-545X

Daniel T. Zhang − *Department of Chemistry, Norwegian University of Science and Technology (NTNU), N-7491 Trondheim, Norway*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jpca.2c06004

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

The authors thank Raffaela Cabriolu and Anders Lervik for their critical comments. They acknowledge funding from the Research Council of Norway through FRINATEK Project No. 275506.

REFERENCES

(1) Metropolis, N.; Rosenbluth, A.; Rosenbluth, M.; Teller, A.; Teller, E. Equation of state calculations by fast computing machines. *J. Chem. Phys.* **1953**, *21*, 1087−1092.

(2) Hastings, W. K. Monte-Carlo Sampling methods using Markov chains and their Applications. *Biometrika* **1970**, *57*, 97−109.

(3) Swendsen, R. H.; Wang, J. S. Replica Monte-Carlo simulation of spin-glasses. *Phys. Rev. Lett.* **1986**, *57*, 2607−2609.

(4) Marinari, E.; Parisi, G. Simulated tempering - a new Monte-Carlo scheme. *Europhys. Lett.* **1992**, *19*, 451−458.

(5) Sugita, Y.; Okamoto, Y. Replica-exchange molecular dynamics method for protein folding. *Chem. Phys. Lett.* **1999**, *314*, 141−151.

(6) Siepmann, J. I.; Frenkel, D. Configurational bias Monte-Carlo - A new sampling scheme for flexible chains. *Mol. Phys.* **1992**, *75*, 59−70.

(7) Vlugt, T. J. H.; Krishna, R.; Smit, B. Molecular simulations of adsorption isotherms for linear and branched alkanes and their mixtures in silicalite. *J. Phys. Chem. B* **1999**, *103*, 1102−1118.

(8) Frenkel, D.; Smit, B.*Understanding Molecular Simulations from Algorithms to Applications*; Academic Press: San Diego, California, U.S.A., 2002.

(9) Dellago, C.; Bolhuis, P. G.; Csajka, F. S.; Chandler, D. Transition path sampling and the calculation of rate constants. *J. Chem. Phys.* **1998**, *108*, 1964.

(10) Swendsen, R. H.; Wang, J.-S. Nonuniversal critical dynamics in Monte Carlo simulations. *Phys. Rev. Lett.* **1987**, *58*, 86−88.

(11) Peters, E. A. J. F.; de With, G. Rejection-free Monte Carlo sampling for general potentials. *Phys. Rev. E* **2012**, *85*, No. 026703.

(12) Michel, M.; Kapfer, S. C.; Krauth, W. Generalized event-chain Monte Carlo: Constructing rejection-free global-balance algorithms from infinitesimal steps. *J. Chem. Phys.* **2014**, *140*, No. 054116.

(13) Plattner, N.; Doll, J. D.; Dupuis, P.; Wang, H.; Liu, Y.; Gubernatis, J. E. An infinite swapping approach to the rare-event sampling problem. *J. Chem. Phys.* **2011**, *135*, 134111.

(14) Plattner, N.; Doll, J. D.; Meuwly, M. Overcoming the Rare Event Sampling Problem in Biological Systems with Infinite Swapping. *J. Chem. Theory Comput.* **2013**, *9*, 4215−4224.

(15) Yu, T.-Q.; Lu, J.; Abrams, C. F.; Vanden-Eijnden, E. Multiscale implementation of infinite-swap replica exchange molecular dynamics. *Proc. Natl. Acad. Sci. U.S.A* **2016**, *113*, 11744−11749.

(16) Lu, J.; Vanden-Eijnden, E. Methodological and Computational Aspects of Parallel Tempering Methods in the Infinite Swapping Limit. *J. Stat. Phys.* **2019**, *174*, 715−733.

(17) Balasubramanian, K.Combinatorics and Diagonals of Matrices. Ph.D. thesis, Loyola College, Madras, India, 1980.

(18) Bax, E.Finite-difference Algorithms for Counting Problems. Ph.D. thesis, California Institute of Technology: Pasadena, United States, 1998.

(19) Bax, E.; Franklin, J.*A finite-Difference Sieve to Compute the Permanent*; CalTech-CS-TR1996; pp 96−04.

(20) Glynn, D. G. The permanent of a square matrix. *Eur. J. Comb.* **2010**, *31*, 1887−1891.

(21) van Erp, T. S. Reaction rate calculation by parallel path swapping. *Phys. Rev. Lett.* **2007**, *98*, No. 268301.

(22) Cabriolu, R.; Refsnes, K. M. S.; Bolhuis, P. G.; van Erp, T. S. Foundations and latest advances in replica exchange transition interface sampling. *J. Chem. Phys.* **2017**, *147*, 152722.

(23) Dellago, C.; Bolhuis, P. G.; Chandler, D. Efficient transition path sampling: Application to Lennard-Jones cluster rearrangements. *J. Chem. Phys.* **1998**, *108*, 9236−9245.

(24) van Erp, T. S.; Moroni, D.; Bolhuis, P. G. A Novel Path Sampling Method for the Sampling of Rate Constants. *J. Chem. Phys.* **2003**, *118*, 7762−7774.

(25) Swenson, D. W. H.; Prinz, J.-H.; Noe, F.; Chodera, J. D.; Bolhuis, P. G. OpenPathSampling: A Python Framework for Path Sampling Simulations. 2. Building and Customizing Path Ensembles and Sample Schemes. *J. Chem. Theory Comput.* **2019**, *15*, 837−856.

(26) Riccardi, E.; Lervik, A.; Roet, S.; Aarøen, O.; van Erp, T. S. PyRETIS 2: An improbability drive for rare events. *J. Comput. Chem.* **2020**, *41*, 370−377.

(27) Riccardi, E.; Dahlen, O.; van Erp, T. S. Fast decorrelating Monte Carlo moves for efficient path sampling. *J. Phys. Chem. Lett.* **2017**, *8*, 4456−4460.

(28) Zhang, D. T.; Riccardi, E.; van Erp, T. S.Enhanced path sampling using subtrajectory Monte Carlo moves. e-Print archive. https://doi.org/10.48550/arXiv.2210.07026.

(29) Au, S.-K.; Beck, J. L. Estimation of small failure probabilities in high dimensions by subset simulation. *Probabilistic Eng. Mech.* **2001**, *16*, 263−277.

(30) Torrie, G.; Valleau, J. Nonphysical sampling distributions in Monte Carlo free-energy estimation: Umbrella samping. *J. Comput. Phys.* **1977**, *23*, 187.

(31) Ghysels, A.; Roet, S.; Davoudi, S.; van Erp, T. S. Exact non-Markovian permeability from rare event simulations. *Phys. Rev. Res.* **2021**, *3*, No. 033068.

(32) Van Erp, T. S.Dynamical Rare Event Simulation Techniques for Equilibrium and Nonequilibrium Systems. In *Advances in Chemical Physics*; John Willey & Sons Inc., 2012; Vol. *151*, p 27.

(33) van Erp, T. S. Efficiency analysis of reaction rate calculation methods using analytical models I: The two-dimensional sharp barrier. *J. Chem. Phys.* **2006**, *125*, 174106.

Supporting Information for: Exchanging Replicas with Unequal Cost, Infinitely and Permanently

Sander Roet[a], Daniel T. Zhang[a], and Titus S. van Erp[a*]

[a]*Department of Chemistry, Norwegian University of Science and Technology (NTNU), N-7491 Trondheim, Norway*

E-mail: titus.van.erp@ntnu.no

1 Supporting Information Text

This Supplementary Information contains the following data, derivations, and numerical examples. In Sec. 2, we describe the complete implementation details for all the ∞RETIS simulations. In Sec. 3, we provide a proof that the replica exchange method with cost unbalanced replicas conserves the equilibrium distribution at the individual ensemble level. Instead of the superstate principle, the derivation is based on the individual ensemble's perspective where the other ensembles serve as an environment, which finally leads to a twisted detailed-balance relation. In Sec.4, we show a $\mathcal{O}(n^2)$ algorithm for computing the P-matrix from a W-matrix for the case that the W-matrix consists of rows having a series with ones, followed by zeros. This is the type of matrix that is relevant for RETIS simulations based on the standard shooting move. Sec. 5 presents the derivations of the theoretical results on the crossing probabilities, rate constant, and permeability via Kramers' theory that are shown in table 1 of the main article. In Sec. 6 the computational efficiencies, including the derivations for the most optimal efficiencies, are discussed. Finally, in Sec. 7 we provide some additional simulation results on the relative transition probabilities through the lower and higher barrier channel.

2 Simulation Methods

The implementation of ∞RETIS was structured as follows. We start $1 \leq K \leq N$ worker- and 1 scheduler-process. Each of the worker-processes is going to process ensemble specific MC moves while the scheduler-process will do all the replica exchange moves and submits new jobs to the workers. All ensembles/trajectories that are currently being updated by a worker-process are not considered for MC moves by the scheduler, essentially being 'locked'. This means that no data is written for those ensembles and they are not valid targets for swapping moves. After a worker is done, it submits the result to the scheduler, the scheduler then unlocks the returned ensemble/trajectory and executes the replica exchange moves on all ensembles/trajectories that that are not locked. It then submits a new job to the freed worker for performing a new MC move in a randomly chosen free ensemble (or two ensembles in case of a point exchange move) and locks the involved ensembles/trajectories.

In the ∞RETIS method there are two kind of ensemble moves that involve MD steps. The first one is the shooting move (either standard shooting[1] or the more recent sub-trajectory moves[2,3]) in which a new path is being generated from an old path within a single ensemble. The second one is the point exchange move between $[0^-]$ and $[0^+]$. If a worker is assigned to this task, it means that both $[0^-]$ and $[0^+]$ are occupied by this worker. The scheduler ensures that there is never more than 1 worker considered free at a given time. When the free worker is assigned to perform a new MC move, each of the ensembles have an equal probability to be selected. If $[0^+]$ or $[0^-]$ is selected and the other is also free, there is a 50% chance to perform a $[0^-] \leftrightarrow [0^+]$ point exchange move instead of a shooting move in the selected ensemble.

Memoryless single variable stochastic (MSVS) process

No actual MD is run for the MSVS simulations. Instead, we directly sample two random numbers, r_1 and r_2 from an uniform distribution $\in [0, 1)$ to set the path's progress and the path length. A path in ensemble $[k^+]$ is assumed to cross interface λ_{k+l} if $r_1 < (0.1)^l$. After this, we wait a random time, $t = 0.2\, r_2\, k + 0.1$ in seconds. This was done to simulate both the increasing average simulation time and variance for outer ensembles. This setup means that we have no history dependence and allows us to compute the theoretical values shown in figure 1. 5 independent ∞RETIS simulations were run with $1, 5, 10, 15, \ldots, 45, 50$ workers.

Two-channel simulations

In order to investigate the effect of our algorithm on the ergodicity of the sampling, a 2D two-channel simulation was run as described in reference[4]. The new RETIS moves introduced in that paper (mirror-move and target-swap move) were not used. Instead, MD was only run to do shooting moves or the $[0^-] \leftrightarrow [0^+]$ point exchanges. As the MD for this system completed too fast, every worker was set to wait 9 times the time it took to run the MD before returning the result. 5 independent ∞RETIS simulations were run with $1, 2, \ldots, 11, 12$ workers.

1D double well with wire fencing

In order to investigate the accuracy with a W matrix that contains more numbers than 0s or 1s we simulated a 1D double-well system[5] together with the high-acceptance version of a novel path-sampling algorithm, wire fencing. The algorithm is described in reference[3], but for us the relevant part is that the high-acceptance weight is the number of frames that a path has outside the interface for each ensemble times an extra factor 2 if the path ends at the last interface. As for the two-channel system, a worker was set to wait 9 times the time it took to complete the MD move before returning the result. 5 independent ∞RETIS simulations were run with $1, 2, \ldots, 7, 8$ workers with interfaces placed at

$[-0.99, -0.8, -0.7, -0.6, -0.5, -0.4, -0.3, 1.0]$.

3 Detailed-balance relations

In this section, we will derive detailed-balance relations for parallel replica's that are not based on the common superstate viewpoint. These alternative relations can be used to validate the replica exchange algorithm for replica's with unequal CPU cost. Our derivation is based on the finite swapping approach, though the infinite swapping version follows automatically from this when the probability to perform a swap goes to unity ($P_{\mathrm{RE}} \to 1$) as explained in the main text. To simplify matters, we assume that we have one type of replica exchange move that is low in CPU cost and one type of ensemble move that operates within one ensemble and has a high CPU cost. The relations that we derive are, however, by no means limited to that. In fact, in the RETIS algorithm there is also a point exchange move between the $[0^-]$ and $[0^+]$ ensemble. In previous publications this move, annotated as $[0^-] \leftrightarrow [0^+]$, was categorized as a special type of swapping/replica exchange move. In this article we reserve the name swap or replica exchange to an operation that involves the swapping of full paths, which does not require any MD steps. In contrast, the $[0^-] \leftrightarrow [0^+]$ point exchange implies the exchange of time slices at the end and start of the paths that are then extended at the other side of the λ_0 interface. In our implementation, this $[0^-] \leftrightarrow [0^+]$ move is carried out by a single worker that locks both the $[0^-]$ and $[0^+]$ ensembles during this move. As the $[0^-]$ paths can never be swapped with any of the other paths, we can view the point exchange move as an ensemble move in ensemble $[0^+]$.

As explained in the main article, the replica exchange algorithm that we propose is based on a set of workers and a set of ensembles. The number of workers K is less than the number of ensembles N. Most of the time the worker is performing a CPU intensive single-ensemble move. The ensemble in which the worker operates is considered occupied/locked. Once a worker has completed a CPU intensive move, the move will be accepted or rejected, after which either a replica exchange move will be carried out with any of the unoccupied ensembles or the worker will be assigned to do a new single-ensemble move at a randomly picked free ensemble.

In order to indicate the difference between occupied and unoccupied ensembles, we introduce a new state vector that indicates both the available ensembles as in the main text and the occupied ensembles with a bar, e. g: $S = (s_1, s_2, \overline{s_3}, s_4, \overline{s_5})$ to show that there are 5 ensembles of which ensemble 3 and 5 are occupied by a worker. For both occupied and unoccupied ensembles, the s_i-terms reflect the most recent state that was sampled in the ith ensemble. Now our sole aim is to ensure that if we just count the instances that an ensemble i is updated with a new sample (which could be a copy of the previous sample in case of a rejected move), these should be distributed according the correct probability density ρ_i.

It is important to note that the time between two updates can vary and depends on the state that was most recently sampled. However, the waiting time between an update of a specific ensemble and the point in time that this ensemble gets occupied by a worker will depend on the states of all other ensembles, but *not* on the state in the ensemble considered. Since the ensembles are independent, this waiting time will be the same on average irrespective to this sampled state. This has as a consequence that if we take "photographs" of the state vector, at intervals or randomly, evenly distributed over time, we should again obtain the correct distributions ρ_i, for all i, of the states in ensemble i as long as we ignore the instances that this ensemble is occupied. In other words, we can write for the previous example state vector

$$\rho(S) = \rho(s_1, s_2, \overline{s_3}, s_4, \overline{s_5}) = \rho_1(s_1)\rho_2(s_2)\rho_3^u(\overline{s_3})\rho_4(s_4)\rho_5^u(\overline{s_5}) \tag{1}$$

where $\rho_i(\cdot)$ is the statistically correct distribution of ensemble i, and $\rho_j^u(\cdot)$ an unknown distribution for occupied ensemble j that has no clear physical interpretation. For instance, it can happen that a state s is relatively unlikely to exist in ensemble i, low $\rho_i(s)$, but that any MC move starting from that state takes a very long time, resulting in a high $\rho_i^u(s)$.

Now, let's consider the Markov chain from the perspective of ensemble 1 where we monitor its state at the point that a new MC is initiated from an old state s_1. From the viewpoint of ensemble 1, the other ensembles are viewed as an "environment" ($\mathcal{E} = (s_2, \overline{s_3}, s_4, \overline{s_5})$ in the aforementioned example), that might or might not influence the MC move. The probability of state s_1 in ensemble 1 can be written as an integral of the conditional probability given an environment:

$$\rho_1(s_1) = \int \rho_1(s_1|\mathcal{E})\rho(\mathcal{E})\mathrm{d}\mathcal{E}. \tag{2}$$

As the ensembles are independent we can write

$$\rho_1(s_1|\mathcal{E}) = \rho_1(s_1), \tag{3}$$

but we temporary keep the condition to clarify the logical structure of the upcoming derivation.

As stated, we assume that we employ two types of moves: 1) a CPU intensive move that modifies s_1 without using the environment $\mathcal{E}$ and 2) a swapping move. In addition, the environment might influence the relative selection probabilities for choosing either 1) or 2). Typically, this selection probability will depend on $N_a(\mathcal{E})$, the number of unoccupied ensembles in $\mathcal{E}$. Further, we need to keep in mind that during the execution of the MC move in ensemble 1, the environment changes. How much the environment changes will depend on how long it takes to fully execute the move involving ensemble 1.

To derive detailed-balance relations for the replica exchange method for cost unbalanced ensembles, we start with the more general balance concept; if we have an infinite number of states distributed according to the equilibrium distribution, all of which make a MC move at the same time, then we have to get the equilibrium distribution again. This means that the flux out off s_1 should be equal to the flux into s_1 which can be written as

$$\int \rho_1(s_1|\mathcal{E})\rho(\mathcal{E})\pi(s_1,\mathcal{E}\to s_1',\mathcal{E}')\mathrm{d}\mathcal{E}\mathrm{d}\mathcal{E}'\mathrm{d}s_1' = \int \rho_1(s_1''|\mathcal{E}'')\rho(\mathcal{E}'')\pi(s_1'',\mathcal{E}''\to s_1,\mathcal{E}''')\mathrm{d}\mathcal{E}''\mathrm{d}\mathcal{E}'''\mathrm{d}s_1'' \tag{4}$$

The transition probability $\pi(\cdot)$ can be split into the transitions via the different types moves (that we will indicate with the Greek letter α) which will be selected with a probability $P_\alpha^{\mathrm{sel}}(\mathcal{E})$ that can depend on the environment $\mathcal{E}$:

$$\pi(s_1,\mathcal{E}\to s_1',\mathcal{E}') = \sum_\alpha P_\alpha^{\mathrm{sel}}(\mathcal{E})\pi_\alpha(s_1,\mathcal{E}\to s_1',\mathcal{E}') \tag{5}$$

This shows another complicating factor as in standard detailed-balance we need to consider the probability that the exact reverse move will be executed once the new state has been established. However, as the environment could have changed, the reverse move might involve different selection probabilities.

By substituting Eq. 5 into Eq. 4, we get an extra summation over α in addition to the integrals:

$$\begin{aligned}&\sum_\alpha \int \rho_1(s_1|\mathcal{E})\rho(\mathcal{E})P_\alpha^{\mathrm{sel}}(\mathcal{E})\pi_\alpha(s_1,\mathcal{E}\to s_1',\mathcal{E}')\mathrm{d}\mathcal{E}\mathrm{d}\mathcal{E}'\mathrm{d}s_1' = \\ &\sum_\alpha \int \rho_1(s_1''|\mathcal{E}'')\rho(\mathcal{E}'')P_\alpha^{\mathrm{sel}}(\mathcal{E}'')\pi_\alpha(s_1'',\mathcal{E}''\to s_1,\mathcal{E}''')\mathrm{d}\mathcal{E}''\mathrm{d}\mathcal{E}'''\mathrm{d}s_1''\end{aligned} \tag{6}$$

But at this point, we apply the first level of "detailedness" by requiring the equation to hold for *each* α:

$$\begin{aligned}&\int \rho_1(s_1|\mathcal{E})\rho(\mathcal{E})P_\alpha^{\mathrm{sel}}(\mathcal{E})\pi_\alpha(s_1,\mathcal{E}\to s_1',\mathcal{E}')\mathrm{d}\mathcal{E}\mathrm{d}\mathcal{E}'\mathrm{d}s_1' = \\ &\int \rho_1(s_1''|\mathcal{E}'')\rho(\mathcal{E}'')P_\alpha^{\mathrm{sel}}(\mathcal{E}'')\pi_\alpha(s_1'',\mathcal{E}''\to s_1,\mathcal{E}''')\mathrm{d}\mathcal{E}''\mathrm{d}\mathcal{E}'''\mathrm{d}s_1''\end{aligned} \tag{7}$$

So now we can evaluate the different moves separately. We further simplify this expression by integration out the variables $\mathcal{E}'$ and $\mathcal{E}'''$ using the following relation:

$$\int \pi_\alpha(s, \mathcal{E} \to s', \mathcal{E}')\mathrm{d}\mathcal{E}' = \pi_\alpha(s, \mathcal{E} \to s', {}^{a}\mathcal{E}) \tag{8}$$

where ${}^{a}\mathcal{E}$ refers to *any* possible environment. Substitution of Eq. 8 in Eq. 7 gives:

$$\int \rho_1(s_1|\mathcal{E})\rho(\mathcal{E})P_\alpha^{\mathrm{sel}}(\mathcal{E})\pi_\alpha(s_1, \mathcal{E} \to s_1', {}^{a}\mathcal{E})\mathrm{d}\mathcal{E}\mathrm{d}s_1' = \int \rho_1(s_1''|\mathcal{E}'')\rho(\mathcal{E}'')P_\alpha^{\mathrm{sel}}(\mathcal{E}'')\pi_\alpha(s_1'', \mathcal{E}'' \to s_1, {}^{a}\mathcal{E})\mathrm{d}\mathcal{E}''\mathrm{d}s_1'' \tag{9}$$

First, we consider $\alpha = 1$ referring the CPU intensive move that only operates in ensemble 1. For this move we substitute $\alpha = 1$ in Eq. 9 and replace $\mathcal{E}''$ and s_1'' with respectively $\mathcal{E}$ and s_1', which is allowed since these are dummy integration variables

$$\int \rho_1(s_1|\mathcal{E})\rho(\mathcal{E})P_1^{\mathrm{sel}}(\mathcal{E})\pi_1(s_1, \mathcal{E} \to s_1', {}^{a}\mathcal{E})\mathrm{d}\mathcal{E}\mathrm{d}s_1' = \int \rho_1(s_1'|\mathcal{E})\rho(\mathcal{E})P_1^{\mathrm{sel}}(\mathcal{E})\pi_1(s_1', \mathcal{E} \to s_1, {}^{a}\mathcal{E})\mathrm{d}\mathcal{E}\mathrm{d}s_1'$$

Then, we fix another level of detailedness by requiring that the integrands at the left and right side of equality sign to be identical for any $\mathcal{E}$ and s_1'. As a result, $\rho(\mathcal{E})P_\alpha^{\mathrm{sel}}(\mathcal{E})$ will cancel out such that we can write

$$\rho(s_1|\mathcal{E})\pi_1(s_1, \mathcal{E} \to s_1', {}^{a}\mathcal{E}) = \rho(s_1'|\mathcal{E})\pi_1(s_1', \mathcal{E} \to s_1, {}^{a}\mathcal{E}) \tag{10}$$

Since in move 1) the ensembles progress independently from each other, we have

$$\pi_1(s_1, \mathcal{E} \to s_1', {}^{a}\mathcal{E}) = \pi_1(s_1 \to s_1')\pi_1(\mathcal{E} \to {}^{a}\mathcal{E}) \tag{11}$$

The subscript "1" in $\pi_1(\mathcal{E} \to {}^{a}\mathcal{E})$ might seem contradictory to the previous statement on independent progression, but it just indicates that the points in time at which the environment is evaluated relates the duration of the MC move in ensemble 1: $\mathcal{E}$ is the environment at the start of the MC move in ensemble 1, and ${}^{a}\mathcal{E}$ is that when the move is completed. As the time for a $s_1 \to s_1'$ move is likely not the same as the time for a $s_1' \to s_1$ move, the final environments are likely not the same. However, ${}^{a}\mathcal{E}$ refers to *any* environment. Hence, by substituting Eq. 11 into Eq. 10, $\pi_1(\mathcal{E} \to {}^{a}\mathcal{E})$ does not only cancel as it appears at both sides of the equals sign, it is also equal to one. We therefore have not just one, but two very good reasons to eliminate this term such that:

$$\rho_1(s_1|\mathcal{E})\pi_1(s_1 \to s_1') = \rho_1(s_1'|\mathcal{E})\pi_1(s_1' \to s_1) \tag{12}$$

or, via Eq. 3:

$$\rho_1(s_1)\pi_1(s_1 \to s_1') = \rho_1(s_1')\pi_1(s_1' \to s_1) \tag{13}$$

This equation essentially the same as the standard detailed balance equation such that we can adapt our acceptance according to

$$P_{\text{acc}}(s_1 \to s_1') = \min\left[1, \frac{\rho_1(s_1')P_{\text{gen}}(s_1' \to s_1)}{\rho_1(s_1)P_{\text{gen}}(s_1 \to s_1')}\right] \tag{14}$$

which is exactly the same as in standard Metropolis-Hastings. Still, the underlying philosophy is different from a super-state perspective as the number of transitions from old to new, $S^{(o)} \to S^{(n)}$, is not the same as from new to old, $S^{(n)} \to S^{(o)}$. Instead, by writing $S = (s_1, \mathcal{E})$ we have that the number of $(s_1^{(o)}, \mathcal{E}^{(o)}) \to (s_1^{(n)}, {}^{a}\mathcal{E}^{(n)})$ transitions should be equal to the number of $(s_1^{(n)}, \mathcal{E}^{(o)}) \to (s_1^{(o)}, {}^{a}\mathcal{E}^{(n)})$ transitions. In addition, as at the end of the move we only update ensemble 1, and not those that are here considered as environment, the number of sampled states in the ensembles do not increase in cohort. Sampling all states simultaneously like in a true superstate move would imply that distributions get mixed with the unknown and unphysical ρ_i^u distributions.

For the swapping move we just consider the example of an attempted $1 \leftrightarrow 2$ swap as all other swaps $i \leftrightarrow j$ are completely analogous. We start again at Eq. 7 with $\alpha = 1 \leftrightarrow 2$, and further we split the environment $\mathcal{E} = \{s_2, \mathcal{E}_{\not 2}\}$ into the part that participates in the swap move, s_2, and the rest, $\mathcal{E}_{\not 2}$:

$$\begin{aligned}
&\int \rho_1(s_1|s_2, \mathcal{E}_{\not 2})\rho_2(s_2)\rho(\mathcal{E}_{\not 2})P^{\mathrm{sel}}_{1\leftrightarrow 2}(\mathcal{E}_{\not 2}) \times \pi_{1\leftrightarrow 2}(s_1, s_2, \mathcal{E}_{\not 2} \rightarrow s_1', s_2', \mathcal{E}_{\not 2}')\mathrm{d}s_2\mathrm{d}\mathcal{E}_{\not 2}\mathrm{d}s_2'\mathrm{d}\mathcal{E}_{\not 2}'\mathrm{d}s_1' = \\
&\int \rho_1(s_1''|s_2'', \mathcal{E}_{\not 2}'')\rho_2(s_2'')\rho(\mathcal{E}_{\not 2}'')P^{\mathrm{sel}}_{1\leftrightarrow 2}(\mathcal{E}_{\not 2}'') \times \pi_{1\leftrightarrow 2}(s_1'', s_2'', \mathcal{E}_{\not 2}'' \rightarrow s_1, s_2''', \mathcal{E}_{\not 2}''')\mathrm{d}s_2''\mathrm{d}\mathcal{E}_{\not 2}''\mathrm{d}s_2'''\mathrm{d}\mathcal{E}_{\not 2}'''\mathrm{d}s_1''
\end{aligned} \tag{15}$$

Here, we assume that the selection probability $P^{\mathrm{sel}}_{1\leftrightarrow 2}$ depends on $\mathcal{E}_{\not 2}$. The chance to do a replica exchange move equals P_{RE}, but once it is decided to perform a replica exchange move, all possible swaps $i \leftrightarrow j$ compete to be selected with an equal probability. Hence, the probability for the $1 \leftrightarrow 2$ swap to be selected depends on the number of available ensembles, which is the total number of ensembles minus the number of occupied ones. This latter information is contained in $\mathcal{E}_{\not 2}$

The swapping transition probability $\pi_{1\leftrightarrow 2}$ relates to a move that has only one possible outcome, namely the one in which the states in ensemble 1 and 2 are exchanged. Therefore, $\pi_{1\leftrightarrow 2}(s_1, s_2, \mathcal{E}_{\not 2} \rightarrow s_1', s_2', \mathcal{E}_{\not 2}')$ is vanishing if $s_1' \neq s_2$ and $s_2' \neq s_1$. Likewise, $\pi_{1\leftrightarrow 2}(s_1'', s_2'', \mathcal{E}_{\not 2}'' \rightarrow s_1, s_2''', \mathcal{E}_{\not 2}''')$ vanishes if $s_2'' \neq s_1$ and $s_1'' \neq s_2'''$. We can, therefore, write

$$\begin{aligned}
\pi_{1\leftrightarrow 2}(s_1, s_2, \mathcal{E}_{\not 2} \rightarrow s_1', s_2', \mathcal{E}_{\not 2}') &= \hat{\pi}_{1\leftrightarrow 2}(s_1, s_2, \mathcal{E}_{\not 2} \rightarrow s_2, s_1, \mathcal{E}_{\not 2}')\delta(s_2 - s_1')\delta(s_1 - s_2') \\
\pi_{1\leftrightarrow 2}(s_1'', s_2'', \mathcal{E}_{\not 2}'' \rightarrow s_1, s_2''', \mathcal{E}_{\not 2}''') &= \hat{\pi}_{1\leftrightarrow 2}(s_2''', s_1, \mathcal{E}_{\not 2}'' \rightarrow s_1, s_2''', \mathcal{E}_{\not 2}''')\delta(s_2''' - s_1'')\delta(s_1 - s_2'')
\end{aligned} \tag{16}$$

where the transition probability with the hat, $\hat{\pi}_{1\leftrightarrow 2}$, differs from transition probability without the hat, $\pi_{1\leftrightarrow 2}$, by the fact that the latter considers any potential (even if impossible) result of the swapping operation, while the former actually relates to the probability of successfully executing the move in practice in which s_1 and s_2 change places. Substitution of Eqs. 16 in Eq. 15 allows us to eliminate the integrals over s_1', s_2', s_1'', and s_2'' via the delta-function integration property.

$$\begin{aligned}&\int \rho_1(s_1|s_2,\mathcal{E}_{\not{2}})\rho_2(s_2)\rho(\mathcal{E}_{\not{2}})P^{\text{sel}}_{1\leftrightarrow 2}(\mathcal{E}_{\not{2}})\hat{\pi}_{1\leftrightarrow 2}(s_1,s_2,\mathcal{E}_{\not{2}}\rightarrow s_2,s_1,\mathcal{E}'_{\not{2}})\mathrm{d}s_2\mathrm{d}\mathcal{E}_{\not{2}}\mathrm{d}\mathcal{E}'_{\not{2}} = \\ &\int \rho_1(s_2'''|s_1,\mathcal{E}''_{\not{2}})\rho_2(s_1)\rho(\mathcal{E}''_{\not{2}})P^{\text{sel}}_{1\leftrightarrow 2}(\mathcal{E}''_{\not{2}})\hat{\pi}_{1\leftrightarrow 2}(s_2''',s_1,\mathcal{E}''_{\not{2}}\rightarrow s_1,s_2''',\mathcal{E}'''_{\not{2}})\mathrm{d}\mathcal{E}''_{\not{2}}\mathrm{d}s_2'''\mathrm{d}\mathcal{E}'''_{\not{2}}\end{aligned} \tag{17}$$

We then eliminate the integrals over $\mathcal{E}'_{\not{2}}$ and $\mathcal{E}'''_{\not{2}}$ using a similar expression as Eq. 8.

$$\begin{aligned}&\int \rho_1(s_1|s_2,\mathcal{E}_{\not{2}})\rho_2(s_2)\rho(\mathcal{E}_{\not{2}})P^{\text{sel}}_{1\leftrightarrow 2}(\mathcal{E}_{\not{2}})\hat{\pi}_{1\leftrightarrow 2}(s_1,s_2,\mathcal{E}_{\not{2}}\rightarrow s_2,s_1,{}^{a}\mathcal{E}_{\not{2}})\mathrm{d}s_2\mathrm{d}\mathcal{E}_{\not{2}} = \\ &\int \rho_1(s_2'''|s_1,\mathcal{E}''_{\not{2}})\rho_2(s_1)\rho(\mathcal{E}''_{\not{2}})P^{\text{sel}}_{1\leftrightarrow 2}(\mathcal{E}''_{\not{2}})\hat{\pi}_{1\leftrightarrow 2}(s_2''',s_1,\mathcal{E}''_{\not{2}}\rightarrow s_1,s_2''',{}^{a}\mathcal{E}_{\not{2}})\mathrm{d}\mathcal{E}''_{\not{2}}\mathrm{d}s_2'''\end{aligned} \tag{18}$$

In the next step, we change some of the dummy integration variable names: s_2''' to s_2 and $\mathcal{E}''_{\not{2}}$ to $\mathcal{E}_{\not{2}}$.

$$\begin{aligned}&\int \rho_1(s_1|s_2,\mathcal{E}_{\not{2}})\rho_2(s_2)\rho(\mathcal{E}_{\not{2}})P^{\text{sel}}_{1\leftrightarrow 2}(\mathcal{E}_{\not{2}})\hat{\pi}_{1\leftrightarrow 2}(s_1,s_2,\mathcal{E}_{\not{2}}\rightarrow s_2,s_1,{}^{a}\mathcal{E}_{\not{2}})\mathrm{d}s_2\mathrm{d}\mathcal{E}_{\not{2}} = \\ &\int \rho_1(s_2|s_1,\mathcal{E}_{\not{2}})\rho_2(s_1)\rho(\mathcal{E}_{\not{2}})P^{\text{sel}}_{1\leftrightarrow 2}(\mathcal{E}_{\not{2}})\hat{\pi}_{1\leftrightarrow 2}(s_2,s_1,\mathcal{E}_{\not{2}}\rightarrow s_1,s_2,{}^{a}\mathcal{E}_{\not{2}})\mathrm{d}\mathcal{E}_{\not{2}}\mathrm{d}s_2\end{aligned} \tag{19}$$

and use a detailed-balance principle by stating that the equality does not only hold when integrated, but is true for any pair $s_2, \mathcal{E}_{\not{2}}$.

$$\rho_1(s_1|s_2,\mathcal{E}_{\not{2}})\rho_2(s_2)\hat{\pi}_{1\leftrightarrow 2}(s_1,s_2,\mathcal{E}_{\not{2}}\rightarrow s_2,s_1,{}^{a}\mathcal{E}_{\not{2}}) = \rho_1(s_2|s_1,\mathcal{E}_{\not{2}})\rho_2(s_1)\hat{\pi}_{1\leftrightarrow 2}(s_2,s_1,\mathcal{E}_{\not{2}}\rightarrow s_1,s_2,{}^{a}\mathcal{E}_{\not{2}}) \tag{20}$$

We further simplify $\rho_1(s_1|s_2,\mathcal{E}_{\not{2}})$ by $\rho_1(s_1)$ using Eq. 3, and split $\hat{\pi}_{1\leftrightarrow 2}(s_1,s_2,\mathcal{E}_{\not{2}}\rightarrow s_2,s_1,{}^{a}\mathcal{E}_{\not{2}})$ into $\hat{\pi}_{1\leftrightarrow 2}(s_1,s_2\rightarrow s_2,s_1)\times\pi_{1\leftrightarrow 2}(\mathcal{E}_{\not{2}}\rightarrow {}^{a}\mathcal{E}_{\not{2}})$ where the latter term cancels like before:

$$\rho_1(s_1)\rho_2(s_2)\hat{\pi}_{1\leftrightarrow 2}(s_1,s_2\rightarrow s_2,s_1) = \rho_1(s_2)\rho_2(s_1)\hat{\pi}_{1\leftrightarrow 2}(s_2,s_1\rightarrow s_1,s_2) \tag{21}$$

Since $\hat{\pi}_{1\leftrightarrow 2}(s_2,s_1\rightarrow s_1,s_2)$ is the transition probability from (s_1,s_2) to (s_2,s_1) in the first two ensembles given that the 1 ↔ 2 swap move was selected, and given that there are no other possible outcomes of this swap ($P_{\mathrm{gen}} = 1$), the transition probability equals the acceptance probability:

$$\rho_1(s_1)\rho_2(s_2)P_{\mathrm{acc}}(s_1, s_2 \rightarrow s_2, s_1) = \rho_1(s_2)\rho_2(s_1)P_{\mathrm{acc}}(s_2, s_1 \rightarrow s_1, s_2) \tag{22}$$

To satisfy this relation, Eq. (4) of the main article suffices.

$$P_{\mathrm{acc}} = \min\left[1, \frac{\rho_1(s_2)\rho_2(s_1)}{\rho_1(s_1)\rho_2(s_2)}\right] \tag{23}$$

So also here, the standard replica exchange acceptance rule applies. The main difference is that ensembles are not updated in cohort. After the 1 ↔ 2 swap move we only update ensembles 1 and 2. Alternatively, after the 1 ↔ 2 swap all other free ensembles will be updated as well with ”null moves”. In the example of Eq. 1 this would mean that besides, ensemble 1 and 2, also ensemble 4 would be updated. As the state in this ensemble is not changing in a 1 ↔ 2 swap, this would imply recounting the existing s_4 state. Hence, this could be viewed as a superstate move, but then without the occupied states. Resampling s_4 is allowed as the chance for resampling is independent of the content of ensemble 4. However, the sampling of the ensembles 3 and 5 should, while occupied, at all cost be avoided since the time that ensembles 3 and 5 remain occupied can correlate with the values of s_3 and s_5, respectively.

Like in Eq. 14, the acceptance rule of Eq. 23 is based on a twisted detailed balance relation: we require that, given an equilibrium distribution, the number of $(s_1^{(o)}, s_2^{(o)}, \mathcal{E}_{\not{1}\not{2}}^{(o)}) \rightarrow (s_1^{(n)}, s_2^{(n)}, {}^{a}\mathcal{E}_{\not{1}\not{2}}^{(n)})$ transitions should be equal to the number of $(s_1^{(n)}, s_2^{(n)}, \mathcal{E}_{\not{1}\not{2}}^{(o)}) \rightarrow (s_1^{(o)}, s_2^{(o)}, {}^{a}\mathcal{E}_{\not{1}\not{2}}^{(n)})$ transitions, where $s_1^{(o)} = s_2^{(n)} = s_1$ and $s_2^{(o)} = s_1^{(n)} = s_2$. So in this section, we proved that standard acceptance-rejection rules can be applied in a parallel scheme in which replica exchange moves occur only between unoccupied ensembles, such that ensembles are not updated in cohort.

4 Matrices with consecutive ones and zeros

If the high-acceptance approach is not applied, $w_i(X) = 1$ in Eq. (6) of the main article and the W-matrix has rows consisting of a sequence of ones, followed a sequence of zeros. The P-matrix can then be determined from Eq. (7) of the main article which has an $\mathcal{O}(n^2)$ scaling. In this section we provide the proof of this equation.

Let n_i be the number of ones in row i. The first step to order the rows with increasing order of n_i. For instance in the following 5×5 matrix

$$W = \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{array} \begin{array}{c} \begin{array}{ccccc} e_1 & e_2 & e_3 & e_4 & e_5 \end{array} \\ \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} \end{array}$$

we see that s_2, originating from an MC move in ensemble e_2, is also valid for e_3 and e_4. State s_3 that was created in e_3 only reaches the minimal condition for that ensemble. In path sampling, where s_2 and s_3 are paths and e_2, e_3 and e_4 refer to path ensembles $[k^+]$, $[l^+]$ and $[m^+]$ with $m > l > k$, it would mean that path s_3 crosses λ_l, but not λ_m, while path s_2 crosses at least $m - k$ more additional interfaces than strictly needed for being a valid trajectory in $e_2 = [k^+]$. As a result, the third row has fewer ones than the second row. After reordering, the W-matrix looks as follows:

$$W = \begin{array}{c|ccccc|} & e_1 & e_2 & e_3 & e_4 & e_5 \\ s'_1 = s_1 & 1 & 1 & 0 & 0 & 0 \\ s'_2 = s_3 & 1 & 1 & 1 & 0 & 0 \\ s_3 = s_2 & 1 & 1 & 1 & 1 & 0 \\ s'_4 = s_4 & 1 & 1 & 1 & 1 & 0 \\ s'_5 = s_5 & 1 & 1 & 1 & 1 & 1 \end{array} = W[n_1, n_2, n_3, n_4, n_5] = W[2, 3, 4, 4, 5]$$

where we introduced the bracket notation $W[\cdot]$ indicating the number of ones in each row in which $1 \leq n_1 \leq n_2 \leq n_3 \ldots \leq n_n = n$. Likewise, we always have $n_i \geq i$.

Based on the recursive relation, $\text{perm}(W) = \sum_j W_{1j}\text{perm}(W\{1j\})$, and the fact that the matrix after removing row 1 and column j, $W\{1j\}$, is identical for any $j \leq n_1$, we can write

$$\text{perm}(W[n_1, n_2, n_3, \ldots, n_n]) = n_1 \times \text{perm}(W[n_2 - 1, n_3 - 1, \ldots, n_n - 1]) \tag{24}$$

The permanent of the remaining matrix $W[n_2 - 1, n_3 - 1, \ldots, n_n - 1]$ can again be written as $(n_2 - 1) \times \text{perm}(W[n_3 - 2, \ldots, n_n - 2])$ and so on. The permanent is, hence, equal to

$$\text{perm}(W[n_1, n_2, \ldots, n_n]) = \prod_{i=1}^{n} (n_i + 1 - i) \tag{25}$$

The P-matrix follows from Eq. (5) of the main article: $P_{ij} = W_{ij}\text{perm}(W\{ij\})/\text{perm}(W)$. This means that $P_{ij} = 0$ whenever $W_{ij} = 0$. If $W_{ij} = 1$, and $n_{i-1} < j$ or $i = 1$, we have that for a matrix $W[n_1, n_2, \ldots, n_{i-1}, n_i, n_{i+1}, \ldots, n_n]$ the following matrix remains after removal of row i and column j:

$$W\{ij\} = W[n_1, n_2, \ldots, n_{i-1}, n_{i+1} - 1, \ldots, n_n - 1] \tag{26}$$

and the permanent

$$\begin{aligned}\mathrm{perm}(W\{ij\}) &= \left(\prod_{i'=1}^{i-1}(n_{i'}+1-i')\right)\left(\prod_{i'=i+1}^{n}(n_{i'}-1+1-(i'-1))\right)\\ &= \left(\prod_{i'=1}^{i-1}(n_{i'}+1-i')\right)\left(\prod_{i'=i+1}^{n}(n_{i'}+1-i')\right) = \frac{\mathrm{perm}(W)}{(n_i+1-i)} \qquad (27)\end{aligned}$$

and, therefore, for this case we have

$$P_{ij} = \frac{1\times \mathrm{perm}(W\{ij\})}{\mathrm{perm}(W)} = \frac{1}{(n_i+1-i)}. \qquad (28)$$

If for some $k < i$, $n_k \geq j$, while $n_{k-1} < j$ or $k = 1$, we have that for a matrix $W[n_1, n_2, \ldots, n_{k-1}, n_k, \ldots, n_i, n_{i+1}, \ldots, n_n]$ the following matrix remains after removal of row i and column j:

$$W\{ij\} = W[n_1, n_2, \ldots, n_{k-1}, n_k-1, n_{k+1}-1, \ldots, n_{i-1}-1, n_{i+1}-1, \ldots, n_n-1] \qquad (29)$$

Therefore, the permanent of $W\{ij\}$ can be written as

$$\begin{aligned}\mathrm{perm}(W\{ij\}) &= \left(\prod_{i'=1}^{k-1}(n_{i'}+1-i')\right)\left(\prod_{i'=k}^{i-1}(n_{i'}-1+1-i')\right)\left(\prod_{i'=i+1}^{n}(n_{i'}+1-1-(i'-1))\right)\\ &= \left(\prod_{i'=1}^{k-1}(n_{i'}+1-i')\right)\left(\prod_{i'=k}^{i-1}(n_{i'}-i')\right)\left(\prod_{i'=i+1}^{n}(n_{i'}+1-i')\right)\\ &= \frac{\mathrm{perm}(W)}{(n_i+1-i)}\prod_{i'=k}^{i-1}\frac{(n_{i'}-i')}{n_{i'}+1-i'} \qquad (30)\end{aligned}$$

This gives for P_{ij}:

$$P_{ij} = \frac{1}{(n_i+1-i)}\prod_{i'=k}^{i-1}\frac{(n_{i'}-i')}{n_{i'}+1-i'} \qquad (31)$$

We can compare this result with that of one row below (row $i+1$):

$$P_{(i+1)j} = \frac{1}{(n_{i+1}+1-(i+1))} \prod_{i'=k}^{i} \frac{(n_{i'}-i')}{n_{i'}+1-i'} = \frac{P_{ij}(n_i+1-i)}{(n_{i+1}-i)} \frac{(n_i-i)}{n_i+1-i} = P_{ij} \frac{n_i-i}{(n_{i+1}-i)} \tag{32}$$

Therefore, we have following recursive relations

$$P_{ij} = \begin{cases} 0, & \text{if } W_{ij}=0 \\ \frac{1}{n_i+1-i}, & \text{if } W_{ij}=1 \text{ and } [W_{(i-1)j}=0 \text{ or } i=1] \\ \left(\frac{n_{i-1}+1-i}{n_i+1-i}\right) P_{(i-1)j}, & \text{otherwise} \end{cases} \tag{33}$$

For the example given above, this relation gives the following P-matrix:

$$P = \begin{array}{r} \\ s_1' = s_1 \\ s_2' = s_3 \\ s_3 = s_2 \\ s_4' = s_4 \\ s_5' = s_5 \end{array} \begin{array}{c} \begin{array}{ccccc} e_1 & e_2 & e_3 & e_4 & e_5 \end{array} \\ \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} & 0 & 0 \\ \frac{1}{8} & \frac{1}{8} & \frac{1}{4} & \frac{1}{2} & 0 \\ \frac{1}{8} & \frac{1}{8} & \frac{1}{4} & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

This $\mathcal{O}(n^2)$ algorithm can be done within a second for $n \leq 3500$, bigger than any foreseeable RETIS simulation, without even leveraging the block-diagonalization. One could swap again the second and third row to get them ordered according to the original s_i-states, though there is in principle no need for this. This is because it is irrelevant to connect the existing states to the ensembles in which they were originally created.

5 Kramers' theory

For Langevin dynamics, Kramers' relation provides a way to improve upon transition state theory via an approximate expression for the transmission coefficient:

$$\kappa = (1/\omega_b)\left(-\gamma/2 + \sqrt{\gamma^2/4 + w_b^2}\right) \tag{34}$$

Here, γ is the friction coefficient of the Langevin dynamics and $\omega_b = \sqrt{k/m}$ with m the particle's mass and k the curvature along the reaction coordinate at the transition state. The rate constant is then the product of the transmission coefficient times the transition state theory expression for the rate:

$$k = \kappa k^{\mathrm{TST}} \tag{35}$$

For a one-dimensional motion along a coordinate z, the transition state theory expression can be expressed as:[6]

$$k^{\mathrm{TST}} = \sqrt{\frac{k_B T}{2\pi m}} \frac{e^{-\beta V(0)}}{\int_{-\infty}^{0} e^{-\beta V(z)} \mathrm{d}z} \tag{36}$$

where $V(\cdot)$ is the underlying potential, T the temperature, k_B the Boltzmann constant, and $\beta = 1/k_B T$. The transition state is here assumed to be located at $z = 0$ and the system is in state A, the reactant state, if $z < 0$.

The Kramers' approximation for the rate constant k follows from Eqs. 34-36. However, other properties like crossing probabilities and the permeability through a membrane can be derived from the transmission coefficient as well.

The crossing probability $P_A(\lambda_B|\lambda_A)$ from interface λ_A to interface λ_B follows from the main TIS/RETIS rate equation:

$$k = f_A P_A(\lambda_B|\lambda_A) \tag{37}$$

where f_A is the conditional flux through λ_A given the system is in state A. Here, λ_A and λ_B correspond to the first, λ_0, and last interface, λ_M, respectively. The flux f_A through λ_A is similar to k^{TST}, the flux through the transition state without recrossing correction, as it counts all positive crossings and is based on the same normalization (integration over state A):

$$f_A = \sqrt{\frac{k_B T}{2\pi m}} \frac{e^{-\beta V(\lambda_A)}}{\int_{-\infty}^{0} e^{-\beta V(z)} \mathrm{d}z} \tag{38}$$

From Eqs. 34-38 we end up with an equation for the crossing probability:

$$P_A(\lambda_B|\lambda_A) = \frac{\kappa e^{-\beta V(0)}}{e^{-\beta V(\lambda_A)}} \tag{39}$$

Hence, based on the underlying potential and Kramers' expression, Eq. 34, one can obtain an approximate value for the crossing probability. Likewise, for a membrane system we can derive a Kramers' expression for the permeability P starting from Eq. 18 in Ref. 4:

$$P = \frac{k}{(\rho_{\mathrm{ref}})_A} = \frac{f_A P_A(\lambda_B|\lambda_A)}{(\rho_{\mathrm{ref}})_A} \tag{40}$$

where ρ_{ref} refers to the probability density for a permeant at a location away from the membrane, z_{ref}, where $V(\cdot)$ is considered to be flat, and the subscript $(\cdot)_A$ indicates that it is normalized over the reactant state region A:

$$(\rho_{\mathrm{ref}})_A = \frac{e^{-\beta V(z_{\mathrm{ref}})}}{\int_{-\infty}^{0} e^{-\beta V(z)} \mathrm{d}z} \tag{41}$$

Note that the integral in the denominator of Eqs. 38 and 41 is usually diverging since the underlying potential $V(\cdot)$ is generally flat away from the barrier in a membrane system. Fortunately, this integral term cancels in Eq. 40:

$$P = \sqrt{\frac{k_BT}{2\pi m}} \left(\frac{e^{-\beta V(\lambda_A)}}{e^{-\beta V(z_{\text{ref}})}} \right) P_A(\lambda_B|\lambda_A) = \sqrt{\frac{k_BT}{2\pi m}} \left(\frac{\kappa e^{-\beta V(0)}}{e^{-\beta V(z_{\text{ref}})}} \right) \tag{42}$$

where in the second equality we substituted $P_A(\lambda_B|\lambda_A)$ using Eq. 39. Hence, based on Eq. 34 and Eq. 42, we can obtain a value for the permeability based on Kramers' theory.

The aforementioned equations can be generalized for multidimensional systems by replacing the $V(z)$ terms with the Landau free energy $F(z)$. That is, for one additional degree of freedom y:

$$F(z) = -k_BT \ln \left(\int e^{-\beta V(y,z)} \mathrm{d}y \right) \tag{43}$$

In addition, if multiple reaction channels yield competing parallel saddle points in the potential energy surface, these need to summed up as we will do in the next section.

5.1 Kramers' relation for crossing probability of a two-channel system

The potential energy surface described in Ref. 4 is the following

$$\begin{aligned} V(y,z) &= e^{-cz^2} \left(V_1 + A + A \sin\left(\frac{2\pi y}{L_y}\right) + B + B\cos\left(\frac{4\pi y}{Ly}\right) \right) \text{ with} \\ &A = (V_2 - V_1)/2,\ B = V_{\text{max}}/2 - V_1/4 - V_2/4, \\ &V_1 = 10,\ V_2 = 11,\ V_{\text{max}} = 20,\ c = 1,\ L_y = 6 \end{aligned} \tag{44}$$

Note that the potential is periodic along the y-direction such that $V(y,z) = V(y+L_y,z)$ and that it is zero in the limit $|z| \to \infty$. Further, the following mass, Langevin friction coefficient and thermodynamic parameters were set in dimensionless reduced units: $\gamma = 5, \quad T = m =$

$k_B = \beta = 1$. The first and last interfaces were set at: $\lambda_A = -1.5$ and $\lambda_B = 1.2$. In this case, we have two saddle points at $(-L_y/4, 0)$ and at $(+L_y/4, 0)$ where the former is slightly lower in potential energy by $1k_BT$ (V_1 and V_2, respectively). The curvatures can be obtained by applying a second order Taylor expansion around $z = 0$:

$$V(-L_y/4, z) \approx V_1 - cV_1z^2 \Rightarrow k_1 = 2cV_1$$
$$V(+L_y/4, z) \approx V_2 - cV_2z^2 \Rightarrow k_2 = 2cV_2$$

which gives $w_{b,1} = \sqrt{20}$ and $w_{b,2} = \sqrt{22}$. As a result $\kappa_1 = 0.5866, \quad \kappa_2 = 0.6002$ via Eq. 34. From this we can compute the crossing probability based on essentially Eq. 39, but using the Landau free energy, $F(\cdot)$, by Eq. 43, instead of the potential energy, $V(\cdot)$, and using both transmission coefficients for the parts along the orthogonal coordinate, y, where they are relevant:

$$P_A(\lambda_B|\lambda_A) \approx \frac{\kappa_1 \int_{-3}^{0} e^{-\beta V(y,0)} \mathrm{d}y + \kappa_2 \int_{0}^{3} e^{-\beta V(y,0)} \mathrm{d}y}{\int_{-3}^{3} e^{-\beta V(y,\lambda_A)} \mathrm{d}y} = 1.61 \cdot 10^{-5} \tag{45}$$

where the integrals over y are taken over one period. Note that the system in Ref. 4 actually contains 3 particles that move in this 2D potential energy surface such that the dimension of the system is actually 6. However, since we follow one single target permeant and the other particles are assumed to have no influence on the target (the interparticle interaction was set to 0[4]), the effective dimension for our analysis is 2 with coordinates y and z.

The permeability then follows from Eq. 42 with $V(\cdot)$ replaced by $F(\cdot)$, where we used the expression based on the crossing probability to have the effect of the two different transmission coefficients directly included:

$$P = \sqrt{\frac{k_BT}{2\pi m}} \left(\frac{\int_{-3}^{3} e^{-\beta V(y,\lambda_A)} \mathrm{d}y}{\int_{-3}^{3} e^{-\beta V(y,z_{\mathrm{ref}})} \mathrm{d}y} \right) P_A(\lambda_B|\lambda_A) = \frac{1}{6}\sqrt{\frac{k_BT}{2\pi m}} \left(\int_{-3}^{3} e^{-\beta V(y,\lambda_A)} \mathrm{d}y \right) P_A(\lambda_B|\lambda_A) = 1.37 \cdot 10^{-6} \tag{46}$$

where we assumed that z_{ref} is taken far away from the membrane at $z = 0$ such that $z_{\mathrm{ref}} \ll 0$ and $V(y, z_{\mathrm{ref}}) \approx 0$.

5.2 Kramers' relation for crossing probability of double well potential

The double well potential is given by[5]

$$V(z) = k_1 z^4 - k_2 z^2 \text{ with } k_1 = 1, \quad k_2 = 2 \tag{47}$$

which has a transition state at $z = 0$ and minima at $z = -1$ and $z = 1$. Further is given that $T = 0.07$ and $k_B = m = 1$ such that the transition state theory expression for the rate, Eq. 36, equals:[5] $k^{\mathrm{TST}} = 2.776 \cdot 10^{-7}$.

The curvature at the transition state equals $2k_2 = 4$ such that $w_b = 2$. Together with the friction coefficient of $\gamma = 0.3$, Kramers' relation, Eq. 34, provides a transmission coefficient: $\kappa = 0.9278$. Henceforth, by Eq. 35 the rate constant based on Kramers' theory equals: $k = 2.58 \cdot 10^{-7}$.

The crossing probability follows from Eq. 39 where in this case $\lambda_A = -0.99$.[3] From the previously determined value for κ, we get: $P_A(\lambda_B|\lambda_A) = 5.83 \cdot 10^{-7}$

6 Computational efficiencies

In this paper, the computational efficiency is defined as

$$\text{efficiency} = \frac{1}{\tau^{\mathrm{eff}}} \tag{48}$$

where τ^{eff} is the efficiency time,[7] which is equal to the computational cost that is needed to get a statistical relative error equal to 1 for the property that is computed. Here, τ^{eff} could be expressed as the number of MD steps in path sampling simulations of large systems or path sampling simulations based on Ab Initio MD where the number of force calculations completely determines the total CPU cost. Expressing the efficiency time is this way has the advantage that it is hardware independent. In this article, however, we express the efficiency time in actual CPU- or wall-time seconds in order to include also the computational cost for calculating the permanents in the replica exchange move.

When a simulation is completed after a certain time τ and the relative error ϵ has been obtained via, e.g. independent runs, block averaging or bootstrapping, the efficiency time is estimated by

$$\tau^{\text{eff}} = \epsilon^2 \tau \tag{49}$$

Note that for serial simulations this property is in principle independent of the simulation length τ. If we run the simulation longer by a certain factor, the error should reduce by the square root of this factor such that τ^{eff} remains unchanged. However, we should realize that there is a rather large statistical uncertainty in the estimated values for τ^{eff} due the fact that the statistical error in the error is generally large.

In the following, unless stated otherwise, we will refer to the CPU-time and CPU-based efficiency time when referring to τ and τ^{eff}. However, let us shortly discuss the wall-time efficiency that follows from the same equation, Eq. 49, but with τ being the wall-time instead of CPU-time. In all our simulations, we fixed the wall-time to 5×12 hours with 5 independent runs. So the wall-time is constant and independent to the number of workers that is used. However, with K workers instead of 1, the CPU-time increases by a factor K. This means that if the error would follow the same trend as in a serial run, the use of K instead of 1 worker would result in a $\sqrt{K}$ reduction of the error. Yet, with τ in Eq. 49 being the wall-time instead of CPU-time, the reduction in the error is not canceled by an increase in τ and the efficiency, Eq. 48, would increase linearly with K. This would mean that we can write:

$$\text{efficiency(wall-time)} = K \times \text{efficiency(CPU-time)} \tag{50}$$

if the parallel run uses the total CPU-time as effectively as a serial simulation that runs $K \times 5 \times 12$ hours long. However, our parallel algorithm will introduce changes in the relative CPU-time that is used for MC moves in the different ensembles. This effect was investigated for the memoryless single variable stochastic (MSVS) process. In the next subsection, we give the meaning and derivation of the continuous curves shown in Fig. 1 of the main article.

6.1 Theoretical efficiencies for the MSVS process

The efficiency time can also be calculated for for specific parts of the calculation. In specific, TIS/RETIS consists of different path ensemble simulations that compute a local crossing probability. In the path ensemble $[k^+]$ which consists of paths that at least cross λ_k, this local crossing probability equals the fraction of paths that cross λ_{k+1} as well. Based on the expected error in the local crossing probability, the CPU-based efficiency time of ensemble $[k^+]$ can be expressed as:[7]

$$\tau_k^{\text{eff}} = \frac{1-p_k}{p_k} \mathcal{N}_k \xi_k L_k \tag{51}$$

where p_k is the local crossing probability of ensemble $[k^+]$, L_k is the average path length (expressed in MD steps or CPU seconds), and ξ_k is the ratio of the average cost of a MC move to L_k. In other words, $\xi_k L_k$ is the average computational cost for doing a MC move (creation of a trial path that might then be accepted or rejected). Finally, $\mathcal{N}_k$ is a measure of the effective correlations between MC moves also called the "statistical inefficiency". Paths can be correlated due to rejections, which implies that the old path is recounted, or because of similarities between accepted paths. In practice, $\mathcal{N}_k$ tends to be significantly larger than 1 while ξ_k is often smaller than 1 as many rejections occur without that a trial path needs to be fully completed. In addition, some MC moves like the replica exchange move or the time-reversal move do not require any MD steps.

In the following, we will neglect the effect that the replica exchange moves have on the errors and on the CPU-time. Under this assumption, the successive MC moves are completely independent. In addition, the ensemble moves are memoryless (hence $\mathcal{N} = 1$). The overall error can thus be computed from the errors in the individual ensembles using standard error propagation rules for independent estimates. Except for the replica exchange part, the MSVS simulation is rejection-free such that we also have $\xi = 1$. In addition, the random artificial MD time for a path in ensemble in ensemble $[k^+]$ was on average $0.1\,k + 0.1$ seconds. To simplify our analysis, we neglect the final 0.1 addition, and state that $L_k = ak$ with $a = 0.1$. Finally, we fixed the local crossing probability to $p_k = p = 1/10$ for all ensembles $[k^+]$ such that

$$\tau_k^{\mathrm{eff}} = a\frac{1-p}{p}k \tag{52}$$

The relative error in estimate of the local crossing probability of ensemble $[k^+]$ follows from Eq. 49:

$$\epsilon_k = \sqrt{\frac{\tau_k^{\mathrm{eff}}}{\tau_k}} \tag{53}$$

with τ_k the CPU-time that is spend to ensemble $[k^+]$. Given a certain division of the total simulation time τ into the times $(\tau_0, \tau_1, \ldots, \tau_{N-1})$, we can compute the total efficiency time by Eq. 49 with

$$\epsilon^2 = \sum_{k=0}^{N-1} \epsilon_k^2 = \sum_{k=0}^{N-1} \frac{\tau_k^{\mathrm{eff}}}{\tau_k} \quad \text{and} \quad \tau = \sum_{k=0}^{N-1} \tau_k \tag{54}$$

The first expression is the standard error propagation rule for the error in a final estimate that is obtained from a product of independent estimates.

Now let us first consider standard TIS or the $N = K$ case. In this simulation we would have an equal number of workers as ensembles. Each worker is solely designated to a single ensemble such that an equal amount of CPU-time is spend per ensemble when the simulation is stopped. So we can simply put $\tau_k = 1$ such that $\tau = N$ and

$$\epsilon^2 = \sum_{k=0}^{N-1} \tau_k^{\text{eff}} = a\frac{1-p}{p}\sum_{k=0}^{N-1} k = a\frac{1-p}{p}\frac{1}{2}(N-1)N \approx \frac{a}{2}\frac{1-p}{p}N^2 \tag{55}$$

where in the last equality we assumed $N \gg 1$. The efficiency time for TIS is hence

$$\tau^{\text{eff}} \approx \frac{1}{2}a\frac{1-p}{p}N^3, \qquad \text{for TIS or } K = N \tag{56}$$

For serial RETIS, each ensemble is updated by a MC move before a next cycle of moves is started. As a result, in each ensemble the same number of MC moves are carried out such that $\tau_k \propto L_k \propto k$. By taking $\tau_k = k$, we get that $\tau = (N-1)N/2 \approx N^2/2$ and

$$\epsilon^2 = \sum_{k=0}^{N-1} \frac{\tau_k^{\text{eff}}}{\tau_k} = aN\frac{1-p}{p} \tag{57}$$

and the CPU-based efficiency time is exactly the same

$$\tau^{\text{eff}} \approx \frac{1}{2}a\frac{1-p}{p}N^3, \qquad \text{for RETIS or } K = 1 \tag{58}$$

This is in agreement with Ref. 7 which stated that an equal division of CPU-time or aiming for the same error in each ensemble gives the same efficiency. Since the local crossing probability is the same for each ensemble, $p_k = p$, aiming for the same error in each ensemble is equivalent to having the same number of MC moves per ensemble (if the statistical inefficiencies, $\mathcal{N}_k$, are the same). The optimal division of CPU-time over the different ensembles is, however, $\tau_k \propto \sqrt{\tau_k^{\text{eff}}}$.[7] By taking $\tau_k = \sqrt{k}$, the total CPU-time becomes

$$\tau = \sum_{k=0}^{N-1} \sqrt{k} \approx \int_0^N \sqrt{x} \mathrm{d}x = \frac{2}{3} N^{3/2} \tag{59}$$

and the total error

$$\epsilon^2 = \sum_{k=0}^{N-1} \frac{\tau_k^{\text{eff}}}{\tau_k} = a\frac{1-p}{p} \sum_{k=0}^{N-1} \sqrt{k} \approx a\frac{1-p}{p}\frac{2}{3} N^{3/2} \tag{60}$$

which by Eq. 49 results in a slightly lower efficiency time than for TIS/RETIS:

$$\tau^{\text{eff}} \approx \frac{4}{9} a \frac{1-p}{p} N^3, \qquad \text{for an optimal division} \tag{61}$$

Based on $a = p = 0.1$ and $N = 50$, the efficiency times are $\tau^{\text{eff}} = 56250$ for TIS/RETIS and $\tau^{\text{eff}} = 50000$ for the optimal division. Naturally, the corresponding CPU-time efficiencies by Eq. 48 are 1/56250 and 1/50000. Furthermore, based on Eq. 50, the optimal wall-time efficiency and the optimal TIS/RETIS wall-time efficiency are given by $K/50000$ and $K/56250$, respectively. These are the continuous black and purple curves in Fig.1d of the main article.

It is interesting to observe that the optimal TIS/RETIS CPU-time efficiency is only 12.5% lower than the optimal CPU-time efficiency. This seems to suggest that it is difficult to improve the CPU-time efficiency of TIS and RETIS unless the division of CPU-time is exactly targeted to do so. On the other hand, one can easily get a much worse CPU-time efficiency when errors in some ensembles are reduced to unnecessary small values while the other ensemble errors are ignored. Based on the fact that $\tau_k \propto \sqrt{\tau_k^{\text{eff}}}$ gives the optimum, the optimum division of MC moves is obtained when in ensembles $[k^+]$ the number of MC moves is proportional to $\sqrt{\tau_k^{\text{eff}}}/L_k$. For the MSVS system this means that the number of executed MC moves in each ensemble should optimally be taken as $\propto 1/\sqrt{k}$ for $k = 1, 2, \ldots, M-1$ (to account for $k = 0$ we should have kept the neglected 0.1 addition in the path length to avoid divergence). This means that it is actually good to execute more MC moves at the lower rank ensembles (low k) than at the higher rank (high k). However, this should not be exaggerated since too many MC moves in the low ranked ensembles will just result in inefficient use of CPU-time as discussed above. Based on the numerical sampling ratios, we determined the CPU-time spend in each ensemble, τ_k, by multiplying these ratios by $L_k = ak$. We then estimated the error based on Eqs. 54 and 52. The resulting efficiency, based on the actual sampling ratios of ∞RETIS, turned out to give a slightly better CPU-time efficiency than that of TIS/RETIS for $15 \leq K \leq 45$, shown by the purple dashed line in Fig. 1d. The resulting wall-time efficiencies of this hybrid theoretical/numerical result is shown by the purple dots if Fig.1d as well. This shows that ∞RETIS can actually improve both the CPU- and wall-time efficiency compared to TIS/RETIS. The latter is expected based on the brute force principle that more CPU power is used per second. The former is more subtle and related to the fact that ∞RETIS leads to a more efficient distribution of the CPU-time among the different ensembles compared to TIS or RETIS.

7 Additional simulation results

7.1 Ratios of channels crossings

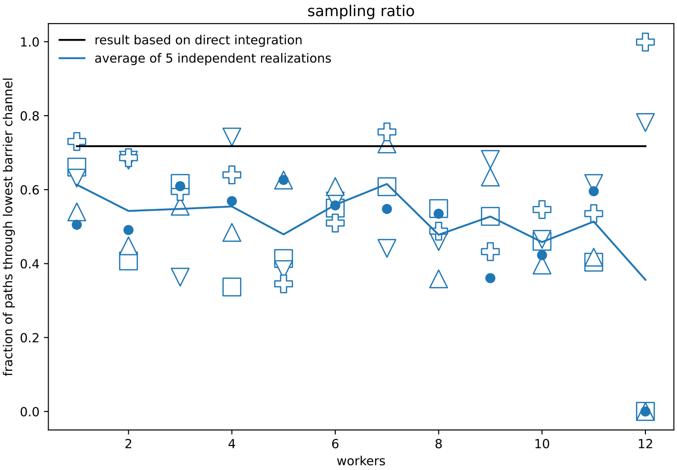


Figure 1: The ratio of first crossings points for the last ensemble in the more favorable channel. The blue icons shows the sampled ratio for each simulation, the blue line is the average of 5 simulations for each amount of workers and the black line is the expected value from direct integration of $\exp(-\beta V(y, z))$ over y with z fixed at $z = \lambda_{10} = -0.2$. This gives an approximate theoretical value of 0.71. The exact value is probably slightly less as the integrated probability density does not correct for phase points that should not be counted as they are actually lying on a path coming from state B rather than A. Still, a majority of simulations provide a fraction that is slightly too low, which is likely an effect of the initial conditions. As reported in Ref. 4, this ratio requires many MC moves to converge without the added MC moves introduced in that paper (mirror-move and target-swap move). These added moves are in principle perfectly compatible with the new replica exchange method, but those were not implemented yet in this work. Therefore, we see the same slow convergence for all of our simulations. Still, the $K < N$ simulations are strikingly better than the $K = N = 12$ case where no replica exchange moves were performed. For 12 workers, the 3 icons overlapping at 0.0 and another simulation showing a fraction equal to 1.0 are the result of the known ergodicity issues of the TIS algorithm due to the lack of swapping moves.

8 Code availability

All simulation code for this paper is publicly available at doi.org/10.5281/zenodo.6977013.

Fair warning; this code is not user-friendly and highly optimized for our specific hardware and output requirements. Instead we would advice everyone to instead use the examples of the infretis github (https://github.com/infretis/infretis).

This code was purely developed to verify the soundness and performance of the algorithm. It uses python multiprocessing with a custom communication code, mimicking MPI, to let the workers run arbitrary python code. It is limited to running on a single machine as no network interface was written for the communication layer and it assumes all files to be accessible by all processes. It uses internal calls to PyRETIS[8] to run the MD for the two-channel and 1D-wirefencing simulations and no current support is present for dealing with external MD engines as with more mature path-sampling codes, like OPS[9] and PyRETIS.[8] It also does not write any data, which means the simulation can not be reanalyzed after it has completed.

A new project has started at https://github.com/infretis/infretis, which will rewrite this custom code to a more user-friendly software. It will add the file handling, use Dask[10] to manage the parallelization which allows for out-of-machine scaling, and will include external MD packages as is common in path sampling codes. Initial examples of the Dask integration can be found in the example directory of that git repository.

References

(1) Dellago, C.; Bolhuis, P. G.; Chandler, D. Efficient transition path sampling: Application to Lennard-Jones cluster rearrangements. *J. Chem. Phys.* **1998**, *108*, 9236–9245.

(2) Riccardi, E.; Dahlen, O.; van Erp, T. S. Fast decorrelating Monte Carlo moves for efficient path sampling. *J. Phys. Chem. Lett.* **2017**, *8*, 4456–4460.

(3) Zhang, D. T.; Riccardi, E.; van Erp, T. S. Enhanced path sampling using subtrajectory Monte Carlo moves. Preprint. 2022; `https://doi.org/10.48550/arXiv.2210.07026`.

(4) Ghysels, A.; Roet, S.; Davoudi, S.; van Erp, T. S. Exact non-Markovian permeability from rare event simulations. *Phys. Rev. Research* **2021**, *3*, 033068.

(5) van Erp, T. S. Dynamical rare event simulation techniques for equilibrium and nonequilibrium systems. *Adv. Chem. Phys.* **2012**, *151*, 27.

(6) Frenkel, D.; Smit, B. *Understanding molecular simulations from algorithms to applications*; Academic press: San Diego, California, U.S.A., 2002.

(7) van Erp, T. S. Efficiency analysis of reaction rate calculation methods using analytical models I: The two-dimensional sharp barrier. *J. Chem. Phys.* **2006**, *125*, 174106.

(8) Riccardi, E.; Lervik, A.; Roet, S.; Aarøen, O.; van Erp, T. S. PyRETIS 2: An improbability drive for rare events. *J. Comput. Chem.* **2020**, *41*, 370–377.

(9) Swenson, D. W. H.; Prinz, J.-H.; Noe, F.; Chodera, J. D.; Bolhuis, P. G. OpenPathSampling: A Python Framework for Path Sampling Simulations. 2. Building and Customizing Path Ensembles and Sample Schemes. *J. Chem. Theory Comput.* **2019**, *15*, 837–856.

(10) Dask Development Team, Dask: Library for dynamic task scheduling. 2016.

PAPER B

The Journal of Chemical Physics

RESEARCH ARTICLE | JANUARY 12 2023

Enhanced path sampling using subtrajectory Monte Carlo moves

Daniel T. Zhang ; Enrico Riccardi ; Titus S. van Erp

J. Chem. Phys. 158, 024113 (2023)
https://doi.org/10.1063/5.0127249

The Journal of Chemical Physics ARTICLE scitation.org/journal/jcp

Enhanced path sampling using subtrajectory Monte Carlo moves

Cite as: J. Chem. Phys. **158**, 024113 (2023); doi: 10.1063/5.0127249
Submitted: 20 September 2022 • Accepted: 20 December 2022 •
Published Online: 12 January 2023

Daniel T. Zhang,[1] Enrico Riccardi,[2] and Titus S. van Erp[1,a)]

AFFILIATIONS

[1] Norwegian University of Science and Technology, Department of Chemistry, NO-7491 Trondheim, Norway
[2] Department of Informatics, UiO, Gaustadalléen 23B, 0373 Oslo, Norway

[a)] Author to whom correspondence should be addressed: titus.van.erp@ntnu.no

ABSTRACT

Path sampling allows the study of rare events, such as chemical reactions, nucleation, and protein folding, via a Monte Carlo (MC) exploration in path space. Instead of configuration points, this method samples short molecular dynamics (MD) trajectories with specific start- and end-conditions. As in configuration MC, its efficiency highly depends on the types of MC moves. Since the last two decades, the central MC move for path sampling has been the so-called shooting move in which a perturbed phase point of the old path is propagated backward and forward in time to generate a new path. Recently, we proposed the subtrajectory moves, stone-skipping (SS) and web-throwing, that are demonstrably more efficient. However, the one-step crossing requirement makes them somewhat more difficult to implement in combination with external MD programs or when the order parameter determination is expensive. In this article, we present strategies to address the issue. The most generic solution is a new member of subtrajectory moves, wire fencing (WF), that is less thrifty than the SS but more versatile. This makes it easier to link path sampling codes with external MD packages and provides a practical solution for cases where the calculation of the order parameter is expensive or not a simple function of geometry. We demonstrate the WF move in a double-well Langevin model, a thin film breaking transition based on classical force fields, and a smaller ruthenium redox reaction at the *ab initio* level in which the order parameter explicitly depends on the electron density.

I. INTRODUCTION

Rare event simulation techniques aim to sample events that require an exceedingly long central processing unit (CPU)/wall time to be simulated with standard molecular dynamics (MD). In classical full atom simulations of protein folding, for example, the longest reported[1] MD runs generated by the special-purpose molecular dynamics Anton 1 supercomputer are around 1 ms, allowing the study of fast-folding proteins. The most recently released Anton 3 supercomputer is even able to generate 100 μs/day for a million-atom system.[2] Despite this remarkable speed, it is still not fast enough to study the folding of all proteins. For instance, the tryptophan synthase β_2 subunit has an experimentally measured[3] folding rate of $k_f = 0.001\ \mathrm{s}^{-1}$. Hence, the protein needs on average 1000 s to fold. The Anton 3 computer would thus need 27 379 wall time years to generate one single transition. For *ab initio* MD (AIMD), the situation is even worse as the quantum mechanical force evaluation is orders of magnitude slower than computing the gradient of a classical force field potential. In addition, no special purpose AIMD computers exist today.

Yet, rare event simulations allow the calculation of rate constants and the study reaction mechanisms orders of magnitude faster than MD, oftentimes without sacrificing any molecular-level resolution.[4] (Replica exchange) transition interface sampling (RE)TIS[5,6] is such a method that exploits the idea of transition path sampling (TPS)[7] to focus the CPU time on the actual barrier crossing event via a Monte Carlo (MC) sampling of MD paths.

RETIS and TIS employ a series of path sampling simulations, each sampling a different path ensemble. The path ensembles differ with respect to a minimal progress requirement, i.e., the number of interfaces (defined by fixed values of the reaction coordinate/order parameter) that has to be crossed.[8] Combining the results of all path ensembles allows the computation of rate constants and other properties with an exponentially reduced CPU time compared to a single MD simulation.

For instance, a classical simulation study on methane hydrate formation[9] using TIS and RETIS reports on a crystallization rate of 10^{-17} nuclei per second per simulation volume. In other words, in a system as small as those used in atomistic simulations, the process for forming a single critical nucleus takes physically 3 years. Naturally, the hypothetical wall time for reaching this with MD is astronomical for any supercomputer. Likewise, RETIS simulations[10] reproduced the rate constant of water dissociation at the AIMD level in reasonable agreement with experiments, suggesting it happens once per 11 h for each water molecule.[11,12] As it required 30 min to produce 1 ps MD time in the 32 water molecules system, a naive straightforward AIMD approach would need 0.7×10^9 centuries of wall time to generate a single dissociation.

Despite being orders of magnitude times faster than plain MD, simulations such as the above are still computationally expensive and can require months to years to obtain satisfactory statistical accuracy. A further increase in efficiency is therefore desirable. There are essentially three approaches to achieve this: (i) reducing the cost of the MC moves, (ii) reducing the number of required trajectories, and (iii) parallelization of the algorithms. Partial path sampling (PPTIS)[13] and milestoning[14] can be viewed as realizations of (i) by sampling more restrictive path ensembles with a reduced average path length. Unfortunately, this introduces additional approximations. Strategies (ii) and (iii), on the other hand, allow for a speed-up while still producing exact results, identical to those from hypothetical unattainably long MD simulations. In fact, RETIS successfully employs strategy (ii) by complementing the shooting moves with replica exchange moves between path ensembles. RETIS is thus more CPU efficient compared to TIS. However, regarding strategy (iii), TIS has the advantage that path ensembles can be run in parallel completely independently, while replica exchange moves require the progress of the sampling in the path ensembles to be synchronized such that processing units do not have to wait for each other. As a result, RETIS might not always outperform TIS based on wall time, which is the reason why the previously mentioned hydrate formation study was partly based on TIS.[9] The recently introduced ∞RETIS algorithm[15] is expected to solve this issue for future studies based on a fundamentally new replica exchange technique for cost-unbalanced replicas.

In fact, ∞RETIS implicitly applies the cost-free replica exchange moves an infinite number of times after each shooting move. Still, replica exchange moves alone are not ergodic and should, therefore, only be used in combination with another MC move like shooting.[16] To further push strategy (ii), the principle MC move should be changed to reduce both the rejection rate and the resemblance between accepted paths. This is exactly what subtrajectory moves aim to establish. These MC moves resemble PPTIS[13] and milestoning[14] in the sense that they evolve via series of shorter paths (subtrajectories/subpaths), but differently to those methods, these subpaths are just intermediates between sampled paths that are extended to their full lengths. Sampled paths, therefore, have no configuration point in common with the previous path, and the statistical inefficiency is typically reduced by a factor equal to the number of intermediate subtrajectories. Hence, while the creation of a full new path becomes more expensive, this is more than offset by the fact that far fewer trajectories are needed to achieve a certain statistical accuracy. In addition, the approach can be combined with a high-acceptance protocol, which minimizes the number of rejections. As a result, most path ensembles obtain a nearly 100% acceptance.[17]

The two moves presented in Ref. 17, stone skipping (SS) and web throwing (WT), however, have one element, the one-step crossing condition, which can hinder the practical implementation with external MD programs or when the calculation of the order parameter is computationally expensive. In SS and WT, the subtrajectories are launched from a configuration point of a previous (sub) path that is just before or after the path ensemble's interface. At this configuration point, velocities are generated such that the interface is crossed again within a single time step. The velocity randomization and one-step crossing test is reiterated several times until the condition is fulfilled. The procedure is based on the idea that generation of new random velocities followed by a one-step crossing test is relatively cheap compared to generating MD steps, especially if the test can be performed without new force calculations. This might not always be the case. Present path sampling codes[18–21] use external MD codes for performing the MD steps. PyRETIS version 2 has, for instance, couplings to Gromacs,[22] Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS),[23] openmm,[24] and CP2K.[25] In order to reduce the number of stop/restart calls to these programs, a "time step" in the RETIS program is often several (10–1000) MD steps by the external MD engine. This complicates the one-step crossing condition as it actually involves not one but several steps, which is costly and not easy to predict without actually performing these steps. Another issue arises when the calculation of the order parameter is expensive, such as those used in nucleation studies.[26,27]

In this article, we discuss several approaches to tackle this issue. The most generically applicable solution is a new member of the subtrajectory family called wire fencing (WF). The approach is slightly more wasteful with respect to the number of MD steps compared to SS but very versatile and does not require any code modifications of the external engines. We illustrate the WF move on three model systems, a simple 1D double-well potential, a Gromacs thin film breakage application, and a CP2K study on ruthenium redox reactions.

II. SUBTRAJECTORY MOVES

The schematic main idea of the three subtrajectory moves is shown in Fig. 1. These are the stone skipping (SS), web throwing (WT), and the new wire fencing (WF) move. The commonality is that an arbitrary number of partial trajectories (subtrajectories/subpaths) are generated before the completion of a new full trajectory. The subtrajectories obey different start- and end-conditions and are, due to this, considerably shorter than full trajectories. The subtrajectories are not part of the sampling but are just intermediate steps between one full trajectory to another. The $[i^+]$ path ensemble that is being sampled in Fig. 1 consists of paths starting at λ_A, crossing λ_i at least once, and ending at either λ_A or λ_B. In Fig. 1, the old full trajectory is colored blue. In the example, the new trajectory is generated via four subtrajectories. The first subtrajectory is obtained from a shooting move from the old trajectory. Then, the next subtrajectory is generated from the previous one until the number of predetermined subtrajectories (4 in this case, colored in orange) is reached. The final subtrajectory is extended backward and forward in time until reaching a stable state. The new full trajectory comprises the last subtrajectory and the extensions colored in green. The

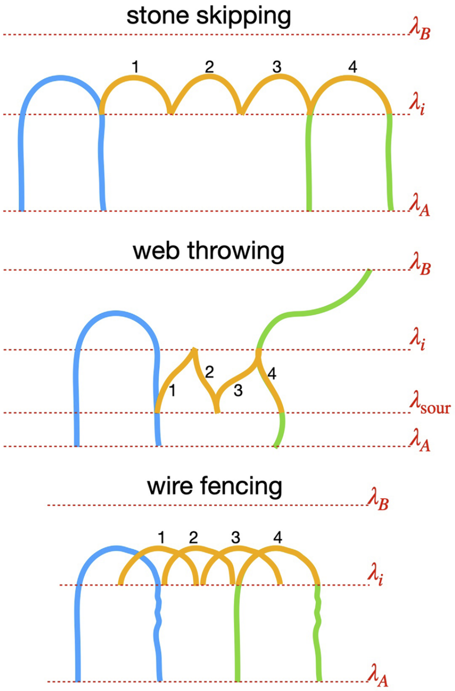


FIG. 1. Cartoon representation of the three subtrajectory moves: stone skipping, web throwing, and wire fencing. The old path is shown in blue. Four subtrajectories are shown in orange. The final new path consists of the fourth subtrajectory and its extensions colored in green.

difference between the three moves lies in the way the shooting of subpaths is executed.

The SS move resembles a flat stone that collides with the water's surface after a skillful throw. The move starts by selecting randomly any of the crossing points of the old path with λ_i, generates new velocities that also establish a crossing, and then proceeds until λ_B is crossed or λ_i is crossed again. The process is then repeated by selecting the subpath's last crossing with λ_i for shooting off the next subpath. Finally, the last subpath is extended and possibly accepted or rejected.[17]

The WT move has been named after a gesture of the famous Marvel character swinging between skyscrapers. Here, an additional interface needs to be defined, *the surface of unlikely return* (SOUR), at the state A side of the λ_i interface. If this interface, λ_{sour}, is crossed toward the direction of state A, it is assumed to be highly unlikely that the MD trajectory will end up in state B rather than A (defined by the last interface, λ_B, and the first interface, λ_A, respectively). The first subpath is then shot from a random crossing point with either λ_i or λ_{sour} at a path segment of the old path that connects these two interfaces. After the velocities of the system's atoms are re-set, like in the SS move, the subpath is continued until λ_{sour} or λ_i is crossed but is only kept if the subpath connects λ_{sour} and λ_i again like the segment of the old path. If not, the subpath is rejected and a new crossing point is taken randomly from the same segment. If both λ_{sour} and λ_i are crossed, the subpath replaces the segment. The process is repeated until the selected number of subpaths, accepted or rejected, has been completed. The final accepted subpath is extended in both time-directions to make a full new path. Note that a rejection of a subpath does not imply a rejection of the MC move itself but just redirects the process of achieving a new path from an old path. The time-direction is chosen such that from λ_{sour}, the trajectory is propagated backward in time and from λ_i forward in time. Due to the placement of λ_{sour}, it is nearly guaranteed that the backward extension reaches state A. As λ_i is also crossed, it is ensured that the path is valid for the $[i^+]$ ensemble, though it might still be rejected due to a final acceptance/rejection step, as required by detailed balance.[28]

The WF move, further discussed in Sec. VII, differs with the other moves by its location of the shooting points. In the WF move, these might be any point with a corresponding value of the reaction coordinate that is larger than λ_i and lower than λ_B (or λ_{cap} if a so-called *cap interface* is set, see Sec. VII). From this point, no specific requirements are needed for the velocities so that they are most conveniently generated from a Maxwell–Boltzmann distribution for the temperature of interest. From the new phase point, MD steps are generated forward and backward in time until λ_B (or λ_{cap}) or λ_i is crossed. The subpath is accepted unless it reaches λ_B (or λ_{cap}) in both time-directions. In that case, it would be rejected and the next shot is taken again from the latest accepted subpath or the previous segment of the old path if no accepted subpaths yet exist. After finishing the number of desired subpaths, the last accepted one is extended to the stable states, like in SS and WT. While the WF move is slightly more wasteful with respect to the MD moves compared to SS, the velocity generation is much simpler, which can have both practical and fundamental advantages compared to SS and WT. These are further discussed in Sec. VI. The name of the WF move is derived from the visual resemblance between the set of full paths and subpaths and the top of a wire fence.

The subtrajectory moves go against strategy (i) as these MC moves require more MD steps than just the number of MD steps for generating a new path. These moves are nevertheless more efficient because they utilize strategy (ii): the statistical inefficiency of the sampling is reduced, and therefore, fewer trajectories are needed to achieve a desired statistical error. Like with the standard shooting move, a final acceptance/rejection step should ensure that the correct statistical distribution of paths is sampled. However, due to the complexity of the subtrajectory move, the design and mathematical validation of the acceptance rule is substantially more complex and is derived from the so-called superdetailed balance[29] principle.

III. SUPERDETAILED BALANCE

The term superdetailed balance was first introduced within the context of configurational bias MC (CBMC),[29–31] which is an effective method to study the adsorption of polymers in nanoporous materials, such as zeolites. In this algorithm, polymers are removed and then regrown atom by atom such that any overlap between the polymer and the zeolite's walls and other polymers is avoided. In this growth process, several attempted branch formations are tested and potentially rejected. Therefore, a specific final accepted configuration could, in principle, be obtained from the old configuration via an infinite number of ways (construction paths). As a result, the Metropolis–Hastings[32] rule for deriving acceptance probabilities becomes impractical as it requires the knowledge on the generation probabilities of all these branches, accepted and rejected, that need to be summed up. This issue is overcome in CBMC using the superdetailed balance principle, which can be formulated in terms of a construction path χ and its inverse $\bar{\chi}$.[17] That is, we not only require detailed balance between any possible old state and new state, but we require this for any specific route that connects these two states,

$$P_{\text{acc}} = \min\left[1, \frac{P(\text{path}^{(n)})P_{\text{gen}}(\text{path}^{(n)} \to \text{path}^{(o)} \text{ via } \bar{\chi})}{P(\text{path}^{(o)})P_{\text{gen}}(\text{path}^{(o)} \to \text{path}^{(n)} \text{ via } \chi)}\right], \tag{1}$$

where $P_{\text{gen}}(\text{path}^{(o)} \to \text{path}^{(n)} \text{ via } \chi)$ is the generation probability to generate the new state (path in our case) from the old state via construction path χ and $P_{\text{gen}}(\text{path}^{(n)} \to \text{path}^{(o)} \text{ via } \bar{\chi})$ is the generation probability to generate the old state from the new state via the reverse construction path $\bar{\chi}$.

In subtrajectory moves, the "construction" path does not only describe the MD extensions of the final path but also the sequence of subtrajectories, including the failed ones. For SS and WT, the unsuccessful velocity generations, that do not obey the one-step crossing condition, should also be considered as part of the construction path χ. In other words, χ consists of several steps, and the generation probability "via χ" is given by the product of generation probabilities of each step.

For each construction path χ, there should exist an unique reverse construction path $\bar{\chi}$. Roughly said, when χ represents a sequence of algorithmic steps, $\bar{\chi}$ will typically consist of the reverse steps in reverse order. However, some groups of consecutive steps might actually happen in the same order. In fact, there is no unique way to define "a reverse," but for a given definition, there will be a one-to-one relation between any possible χ and its reverse $\bar{\chi}$ and with that, valid acceptance/rejection rules can be derived based on the superdetailed balance [Eq. (1)].

Yet, the definition of the reverse should be chosen such that the acceptance probability is computable and not negligibly small in the majority of cases. Therefore, the mathematical definition for the inverse is taken such that the probabilities of most of the algorithmic steps in the expressions for $P_{\text{gen}}(\text{path}^{(o)} \to \text{path}^{(n)} \text{ via } \chi)$ and $P_{\text{gen}}(\text{path}^{(n)} \to \text{path}^{(o)} \text{ via } \bar{\chi})$ will cancel.

For instance, if we represent the construction path as a vector containing the different steps in chronological order, χ could look like

$$\chi = [s^0, t^1, t^2, s^3, s^4, t^5, s^6], \tag{2}$$

which shows that there were six subtrajectories generated of which there were three failed trials t^1, t^2, and t^5. The initial step involves cutting out the very first subtrajectory s^0 from the old path, while the final step implies not only the generation of the last subtrajectory s^6 but also its extension to a full trajectory. The reverse construction path in this case is conveniently defined as

$$\bar{\chi} = [s^6, s^4, t^5, s^3, s^0, t^1, t^2]. \tag{3}$$

Hence, the order of the steps is not completely reversed, but the reverse order takes place on groups of consecutive steps, a group being a successful subtrajectory with all its failed trials that follow. The reason for this inverse is that Eq. (2) shows that trial trajectory t^5 can be generated starting from s^4, but this is not necessarily the case from s^6. Reversely, as s^6 was generated from s^4, they share a common configuration point, which makes it possible to generate s^4 from s^6. There is, however, no reason whatsoever that s^6 and t^5 share a common configuration point. Hence, if we would consider the reverse to be $\bar{\chi} = [s^6, t^5, s^4, \ldots]$, $P_{\text{gen}}(\text{path}^{(n)} \to \text{path}^{(o)} \text{ via } \bar{\chi})$ would most likely be zero as $\bar{\chi}$ itself cannot be generated. In contrast, the inverse based on the grouped reordering, Eq. (3), contains generation probabilities, such as the probability to generate t^5, given s^4, which appear both in χ and $\bar{\chi}$. Therefore, all the generation probabilities of failed trajectories cancel in Eq. (1). Likewise, all failed velocity generations in SS and WT that do not obey the one-step crossing condition cancel out for the same reason, as shown in Ref. 17.

Excluding all the failed steps that will cancel in Eq. (1), we can write for $P_{\text{gen}}(\text{path}^{(o)} \to \text{path}^{(n)} \text{ via } \chi)$,

$$\begin{aligned} P_{\text{gen}}\Big(\text{path}^{(o)} &\to \text{path}^{(n)} \text{ via } \chi\Big) \propto P_{\text{sel}}(s^0|\text{path}^{(o)}) \\ &\times P_{\text{sel}}(r^{0,3}|s^0)P_{\text{gen}}(v^{0,3})P_{\text{MD}}(s^3|x^{0,3}) \\ &\times P_{\text{sel}}(r^{3,4}|s^3)P_{\text{gen}}(v^{3,4})P_{\text{MD}}(s^4|x^{3,4}) \\ &\times P_{\text{sel}}(r^{4,6}|s^4)P_{\text{gen}}(v^{4,6})P_{\text{MD}}(s^6|x^{4,6}) \\ &\times P_{\text{sel}}(\text{td})P_{\text{MD}}(\text{path}^{(n)}|s^6). \end{aligned} \tag{4}$$

Here, $P_{\text{sel}}(s^0|\text{path}^{(o)})$ is the probability for selecting s^0 from the old path$^{(o)}$ and $P_{\text{sel}}(r^{0,3}|s^0)$ is the selection probability of choosing point $r^{0,3}$ from the subpath s^0 as the shooting point. Since $r^{0,3}$ is a shooting point to go from s^0 to s^3, it is a configuration point that s^0 and s^3 have in common. $P_{\text{gen}}(v^{0,3})$ is the probability for generating the velocities $v^{0,3}$, which are the velocities of s^3 at the corresponding configuration point $r^{0,3}$. $P_{\text{MD}}(s^3|x^{0,3})$ is the chance that starting from phase point $x^{0,3} = (r^{0,3}, v^{0,3})$, the MD integrator produces subpath s^3 by integrating the equations of motion forward and backward in time. The MD integrator can be based on actual Newtonian MD, Langevin, Brownian, etc. Likewise, $P_{\text{MD}}(\text{path}^{(n)}|s^6)$ is the chance that the new path$^{(n)}$ is produced by extending the final subpath s^6. Finally, $P_{\text{sel}}(\text{td})$ is the selection probability for the time-direction along the new path. Note that the time-direction along the subpaths is irrelevant in WT and WF. In SS, subpaths do have a sort of direction as the next shooting always takes place at the last λ_i crossing.[17]

For the reverse construction path [Eq. (3)], we can write

$$\begin{aligned} P_{\text{gen}}\Big(\text{path}^{(n)} &\to \text{path}^{(o)} \text{ via } \bar{\chi}\Big) \propto P_{\text{sel}}(s^6|\text{path}^{(n)}) \\ &\times P_{\text{sel}}(r^{6,4}|s^6)P_{\text{gen}}(v^{6,4})P_{\text{MD}}(s^4|x^{6,4}) \\ &\times P_{\text{sel}}(r^{4,3}|s^4)P_{\text{gen}}(v^{4,3})P_{\text{MD}}(s^3|x^{4,3}) \\ &\times P_{\text{sel}}(r^{3,0}|s^3)P_{\text{gen}}(v^{3,0})P_{\text{MD}}(s^0|x^{3,0}) \\ &\times P_{\text{sel}}(\text{td})P_{\text{MD}}(\text{path}^{(o)}|s^0). \end{aligned} \tag{5}$$

Now, it becomes apparent that most terms will cancel out in Eq. (1) when we take the ratio between Eqs. (5) and (4). First of all, the time-direction is chosen with a 50% probability such that $P_{\text{sel}}(\text{td}) = 0.5$. Then, we can use the fact that a path probability can be written in terms of a phase point probability times the MD generation probability,

$$P(\text{path}) = \rho(x) P_{\text{MD}}(\text{path}|x), \tag{6}$$

where $\rho(x)$ is the phase space equilibrium density for any phase point x that is part of the path.[17] For a phase point $x = (r, v)$, this can be split into

$$\rho(x) = \rho_r(r)\rho_v(v), \tag{7}$$

where ρ_r and ρ_v are, respectively, the configuration (Boltzmann) distribution and the velocity Maxwell–Boltzmann distribution (possibly subjected to bond- and angle constraints if applicable). Furthermore, as generating new velocities in Eqs. (4) and (5) is based on the velocity distribution, $P_{\text{gen}}(v) = \rho_v(v)$, we can substitute all P_{MD} terms in Eqs. (4) and (5), e.g.,

$$\begin{aligned} P_{\text{gen}}(v^{4,6}) P_{\text{MD}}(s^6|x^{4,6}) &= \frac{P_{\text{gen}}(v^{4,6}) P(s^6)}{\rho(x^{4,6})} \\ &= \frac{\rho_v(v^{4,6}) P(s^6)}{\rho(x^{4,6})} = \frac{P(s^6)}{\rho_r(r^{4,6})}, \\ P(s^6) P_{\text{MD}}(\text{path}^{(n)}|s^6) &= P(\text{path}^{(n)}). \end{aligned} \tag{8}$$

Applying these operations to Eqs. (4) and (5), we get

$$\begin{aligned} P_{\text{gen}}\Big(\text{path}^{(o)} \to \text{path}^{(n)} \text{ via } \chi\Big) &\propto P_{\text{sel}}(s^0|\text{path}^{(o)}) P_{\text{sel}}(\text{td}) \\ &\times P_{\text{sel}}(r^{0,3}|s^0) P_{\text{sel}}(r^{3,4}|s^3) P_{\text{sel}}(r^{4,6}|s^4) \\ &\times P(s^3) P(s^4) P(\text{path}^{(n)}) / [\rho_r(r^{0,3}) \rho_r(r^{3,4}) \rho_r(r^{4,6})], \\ P_{\text{gen}}\Big(\text{path}^{(n)} \to \text{path}^{(o)} \text{ via } \overline{\chi}\Big) &\propto P_{\text{sel}}(s^6|\text{path}^{(n)}) P_{\text{sel}}(\text{td}) \\ &\times P_{\text{sel}}(r^{6,4}|s^6) P_{\text{sel}}(r^{4,3}|s^4) P_{\text{sel}}(r^{3,0}|s^3) \\ &\times P(s^4) P(s^3) P(\text{path}^{(o)}) / [\rho_r(r^{6,4}) \rho_r(r^{4,3}) \rho_r(r^{3,0})]. \end{aligned} \tag{9}$$

In the ratio of these two equations, more terms will cancel out as $r^{\alpha,\beta} = r^{\beta,\alpha}$. Furthermore, $P_{\text{sel}}(\text{td}) = 0.5$ as stated before. In all subtrajectory moves, $P_{\text{sel}}(r|s^\alpha)$ is either a fixed number (SS and WT) or it depends on s^α, but not on r (WF). In SS, the shooting point is selected from the last crossing with λ_i, and therefore, $P_{\text{sel}}(r|s^\alpha) = 2$ (the phase point just before or after λ_i). In WT, it is randomly chosen from a crossing with either λ_i or λ_{sour}, and therefore, $P_{\text{sel}}(r|s^\alpha) = 4$. With stochastic dynamics, one can also opt to choose only the inner points[17] such that $P_{\text{sel}}(r|s^\alpha) = 2$. In WF, any point of the subpath that lies between λ_i and λ_B (or λ_{cap}) can be chosen. In all these cases, the $P_{\text{sel}}(r|s^\alpha)$ terms with identical s^α cancel out in the ratio. That means that the only terms that remain depend on the first and last subpath (s^0 and s^6) or on the full paths ($\text{path}^{(o)}$ and $\text{path}^{(n)}$),

$$\begin{aligned} &\frac{P_{\text{gen}}(\text{path}^{(n)} \to \text{path}^{(o)} \text{ via } \overline{\chi})}{P_{\text{gen}}(\text{path}^{(o)} \to \text{path}^{(n)} \text{ via } \chi)} \\ &\quad = \frac{P_{\text{sel}}(s^6|\text{path}^{(n)}) P_{\text{sel}}(r^{6,4}|s^6) P(\text{path}^{(o)})}{P_{\text{sel}}(s^0|\text{path}^{(o)}) P_{\text{sel}}(r^{0,3}|s^0) P(\text{path}^{(n)})} \\ &\quad = \frac{P_{\text{sel}}(r^{6,4}|\text{path}^{(n)}) P(\text{path}^{(o)})}{P_{\text{sel}}(r^{0,3}|\text{path}^{(o)}) P(\text{path}^{(n)})} = \frac{P(\text{path}^{(o)})/M^{(n)}}{P(\text{path}^{(n)})/M^{(o)}}, \end{aligned} \tag{10}$$

where in the third expression, we contracted the selection probabilities involving the two-steps (first selecting s^0 or s^6 and then selecting $r^{0,3}$ or $r^{6,4}$) to the chance of selecting the very first successful crossing point from the existing full path. Finally, the latter was replaced by $1/M^{(n)}$ and $1/M^{(o)}$, where $M^{(n)}$ and $M^{(o)}$ are the numbers of different equally probable possibilities to select a shooting point for generating a subtrajectory from the new and old full path, respectively.

If we substitute Eq. (10) into Eq. (1), we obtain a rather simple expression for the acceptance,

$$P_{\text{acc}} = \min\left[1, \frac{M^{(o)}}{M^{(n)}}\right]. \tag{11}$$

In SS, $M^{(o)}$ and $M^{(n)}$ are simply proportional to the number of crossing points of the old and new paths with λ_i, while for WT, these are proportional to the number segments that can be cut out of these trajectories that connect λ_{sour} and λ_i.[17] In WF, these relate to the number of points between λ_i and λ_B. If a so-called *cap*-interface is defined, $M^{(o)}$ and $M^{(n)}$ relate to the number of points between λ_i and λ_{cap} excluding any points lying on a segment $\lambda_{\text{cap}} \to \lambda_{\text{cap}}$ without crossing λ_i.

Equation (11) can also be combined with an early rejection scheme as was introduced in Ref. 5. In the standard approach, one would complete the MC move, compute the acceptance probability [Eq. (11)], take a uniform random number α between 0 and 1, and then accept if $\alpha < P_{\text{acc}}$ and reject otherwise. In the early rejection scheme, the random number α is taken first and the move is rejected as soon as $M^{(n)} > M^{(o)}/\alpha$. In normal shooting, this provides a considerable speed up since long paths have a high chance to get rejected. Using the early rejection scheme, a lot of unnecessary MD steps can be avoided as these paths can be stopped whenever they exceed the predetermined maximum length. Yet, for the subtrajectory moves, the *high-acceptance* scheme is preferable as we discuss in Sec. V. In Sec. IV, we show why the subtrajectory moves allow us to sample fewer trajectories than with standard shooting via a reduction of the statistical inefficiency.

IV. STATISTICAL INEFFICIENCY

The principal property that is computed in the $[i^+]$ ensemble is the local crossing probability $P_A(\lambda_{i+1}|\lambda_i)$. This is the history dependent conditional probability that the system, given it crosses λ_A and then crosses λ_i, crosses λ_{i+1} before λ_A. In the *post hoc* analysis, this local crossing probability is simply the fraction of sampled path in the $[i^+]$ ensemble that happen to cross λ_{i+1} in addition to λ_i. Once these are accurately enough determined, the global crossing probability $P_A(\lambda_B|\lambda_A)$ is obtained from[5,8]

$$P_A(\lambda_B|\lambda_A) = \prod_{i=0}^{n-1} P_A(\lambda_{i+1}|\lambda_i), \tag{12}$$

where $\lambda_0 = \lambda_A$ and $\lambda_n = \lambda_B$. The above expression is exact since the local crossing probabilities include the full history dependence ($\lambda_A \to \lambda_i$) in their condition.[33] An alternative approximate expression for the global crossing probability is used in partial path TIS[13] in which the amount of spatial memory is reduced though not set to zero, as in milestoning.[14] The global crossing probability gives the rate of the transition when multiplied with f_A, the conditional flux through λ_A.

In TIS, the flux is calculated by straightforward MD where the system is prepared in state A and then the number of crossings with λ_A per time unit is computed. If a spontaneous transition to state B takes place, which is unlikely for a rare event, the simulation is paused, reinitiated in state A and then continued. RETIS computes the flux term differently as it does not use a single continuous MD simulation. Instead, the $[0^-]$ path ensemble is introduced to explore the A state, and the flux is derived from the average path lengths in $[0^-]$ and $[0^+]$.[6] In addition to rate constants, the overall crossing probability can also be used to compute permeability coefficients[34] and activation energies.[35,36]

Considering the jth path in the simulation for path ensemble $[i^+]$, the main output of sample j (the generated path) that is relevant for the computation of the crossing probability is simply the observation of whether it crosses λ_{i+1} or not. We can describe this by a characteristic function h_j, which equals 1 if λ_{i+1} is crossed and 0 otherwise. The simulation estimate of the local crossing probability, $p(m)$, after m MC moves is then expressed as

$$p(m) = \frac{1}{m}\sum_{j=0}^{m-1} h_j \approx P_A(\lambda_{i+1}|\lambda_i), \tag{13}$$

where the index counter starts from zero for mathematical convenience.

For finite m, the value of $p(m)$ will not be exact, and the absolute error, ϵ_a, is defined as the standard deviation of the mean $\sigma_{p(m)}$. This is essentially the standard deviation in possible $p(m)$ results if the simulation experiment would be carried out multiple times. Mathematically, we can write this as

$$\epsilon_a = \sigma_{p(m)} = \sqrt{\langle (p(m) - p)^2\rangle}, \tag{14}$$

where $p = p(\infty) = P_A(\lambda_{i+1}|\lambda_i)$ and the brackets $\langle\cdot\rangle$ refer to the perfect ensemble sampling average. This can be viewed as the hypothetical average that is obtained after repeating the simulation an infinite number of times starting with initial conditions that are randomly drawn form a perfect statistical equilibrium distribution. In other words, we have $\langle p(1)\rangle = \langle p(m)\rangle = p$. Furthermore, since detailed balance MC moves conserve the equilibrium distribution,[29] the absolute value of the index j is irrelevant and $\langle h_0\rangle = \langle h_1\rangle = \langle h_j\rangle = p$ and $\langle h_j h_k\rangle = \langle h_0 h_{k-j}\rangle$ for any j, k. Using this, one can show that[37]

$$\sigma^2_{p(m)} = \frac{\sigma^2_{p(1)}}{m}\mathcal{N}, \quad \mathcal{N} = [1 + 2n_c], \tag{15}$$

where $\mathcal{N}$ is called the statistical inefficiency and n_c is the correlation number, which is the integral of the correlation function $C(j)$,

$$n_c = \sum_{j=1}^{\infty} C(j), \quad C(j) = \frac{\langle (h_0 - p)(h_j - p)\rangle}{\langle (h_0 - p)^2\rangle}. \tag{16}$$

As the output h_j of a single sample is either 1 with a probability p or 0 with a probability $(1 - p)$, the sample standard deviation $\sigma_{p(1)}$ can be simplified as

$$\begin{aligned}\sigma^2_{p(1)} &= \langle (p(1) - p)^2\rangle = \langle (h_0 - p)^2\rangle \\ &= p(1 - p)^2 + (1 - p)(0 - p)^2 = p(1 - p).\end{aligned} \tag{17}$$

Using Eqs. (15)–(17), we can write for the relative error,

$$\epsilon_r = \frac{\epsilon_a}{p} = \sqrt{\frac{1 - p}{p}\frac{\mathcal{N}}{m}}. \tag{18}$$

Equation (18) shows that for a fixed number of MC moves m, the larger the local crossing probability $p = P_A(\lambda_{i+1}|\lambda_i)$, the lower the relative error. Hence, the result in simulation $[i^+]$ converges faster when the difference between λ_i and λ_{i+1} is small, but this will obviously increase the number of path ensembles needed. Analytical results on model systems suggest that the optimum placement of interfaces in TIS is achieved when $p \approx 0.2$ for all ensembles.[37] In RETIS, the optimum is expected to be slightly higher as this would lead to more successful swaps. Likewise, the optimum is also slightly higher if the weighted histogram analysis method (WHAM)[38] is used instead of single-point matching to determine the total crossing probability. In this approach, the crossing statistics of the path ensemble $[i^+]$ is not limited to the fraction of paths crossing λ_{i+1}, but also the fractions for crossing λ_{i+2}, λ_{i+3}, etc., are used to get a slightly more accurate estimate of Eq. (12).[39,40]

If the sampling between successive MC moves is completely uncorrelated, we have that $\langle (h_0 - p)(h_j - p)\rangle = \langle (h_0 - p)\rangle \cdot \langle (h_j - p)\rangle = 0 \cdot 0 = 0$. This would imply that $C(j) = n_c = 0$ and $\mathcal{N} = 1$. In this case, if $p = 0.2$, there are about $m = 400$ trajectories required to obtain an $\epsilon_r = 10\%$ error. For $\mathcal{N} > 1$, one would need $m/\mathcal{N} = m_u = 400$ to get the same error. Here, m_u is called the number of effectively uncorrelated samples.

In general, $C(j) \neq 0$ except for the limit $j \to \infty$ as correlation decays. If a MC move is rejected at step j, then the previous sample is kept and recounted such that sample j is identical to sample $j - 1$. Hence, if there are j consecutive rejections, sample j is identical to sample 0 causing correlation over multiple steps. Even if the jth step is accepted, it tends to have some similarity with the previous sample. Therefore, there is a high probability for $h_j = h_{j-1}$, even if the samples are not identical. The correlations lead to a sampling output $(h_0, h_1, h_2, \ldots)$ with long rows of consecutive zeros and consecutive ones.

To illustrate this effect with a mathematical example, suppose that the MC move has a probability π_R to remain unchanged such that $h_j = h_{j-1}$ and a probability $\pi_M = 1 - \pi_R$ to actually make a move that potentially (but not necessarily) changes the output: the new sample yields $h_j = 1$ with a probability p and $h_j = 0$ with a probability $(1 - p)$. As shown in the Appendix, for this mathematical model, the statistical inefficiency equals

$$\mathcal{N} = \frac{2 - \pi_M}{\pi_M}. \tag{19}$$

This shows that for a typical MC acceptance probability of 50%, the effect of rejections alone causes the statistical inefficiency to be equal to 3. The situation is usually worse in complex systems and also more

16 July 2023 08:56:40

difficult to identify than merely by the presence of rows of consecutive ones or zeros. For instance, inter- and intramolecular changes of reactants could temporarily boost or reduce the probability of a transition. The same kind of fluctuations in the temporary transition probability can be caused by the local solvent structure and the position and orientation of catalytic molecules. These describe degrees of freedom that are orthogonal to the reaction coordinate.

We can examine this by a slightly more complex model where we assume that there are two phases α and β, described by the orthogonal degrees of freedom, which occur with probabilities P_α and $P_\beta = 1 - P_\alpha$. Let p_α and p_β be the corresponding local crossing probabilities along the reaction coordinate for these phases such that $p = P_\alpha p_\alpha + P_\beta p_\beta$. Analogous to the above, let π_ρ be the chance to not update the phase, and $\pi_\mu = 1 - \pi_\rho$ be the chance to freshly choose between phase α or β with respective probabilities P_α and P_β. As shown in the Appendix, in this case, the statistical inefficiency equals

$$\mathcal{N} = \frac{2K_s - \pi_\mu(2K_s - 1)}{\pi_\mu}, \tag{20}$$

where K_s is a system parameter that does not depend on the type of MC move,

$$K_s = \frac{P_\alpha P_\beta (p_\alpha - p_\beta)^2}{p(1-p)} = \frac{(p - p_\alpha)(p_\beta - p)}{p(1-p)}. \tag{21}$$

Note that $K_s = 0$ whenever $p_\alpha = p_\beta$, which gives $\mathcal{N} = 1$. This would be the case if all TIS interfaces are placed at isocommittor surfaces, which partly supports the hypothesis of Ref. 41 that stated that path sampling simulations are most efficient if the reaction coordinate λ equals the committor. However, although this surely minimizes the statistical inefficiencies, the mean path lengths in the path ensembles also depend on the choice of the reaction coordinate λ. If this is included in the analysis, the hypothesis is at least not generally true.[33]

Now, assume that not all generated paths are saved and analyzed, but instead only every N_sth path is kept. While this will cause a reduction in the number of samples from m to m/N_s, it does not necessarily reduce the number of uncorrelated samples m_u as the statistical inefficiency between saved samples is also reduced. In particular, the "remain" probability between saved samples changes from π_ρ to $\pi_\rho^{N_s}$ and, therefore, the "move" probability changes from π_μ to $1 - \pi_\rho^{N_s} = 1 - (1 - \pi_\mu)^{N_s}$. The statistical inefficiency between saved samples is henceforth

$$\mathcal{N}(N_s) = \frac{2K_s - (1 - (1 - \pi_\mu)^{N_s})(2K_s - 1)}{1 - (1 - \pi_\mu)^{N_s}}. \tag{22}$$

Equation (22) shows that the statistical efficiency indeed goes down with increasing N_s up to an asymptote equal to 1. Taking the power series up to first order in π_μ, we see that the initial downfall is inversely linear,

$$\mathcal{N}(N_s) \approx \frac{2K_s - N_s\pi_\mu(2K_s - 1)}{N_s\pi_\mu} \approx \frac{\mathcal{N}(1)}{N_s}, \tag{23}$$

where we assumed $N_s\pi_\mu \ll 1$. As a result, saving every N_sth path instead of all paths will not affect much the post-simulation analysis in terms of accuracy. The reduction in the number of data points from m to m/N_s is compensated by a lower statistical inefficiency such that the number of uncorrelated samples m_u remains nearly unchanged. While this allows for obvious data storage savings, reducing both the memory and time for writing to disk, it also paves the way to reduce MD steps, as shown in Fig. 2. The figure

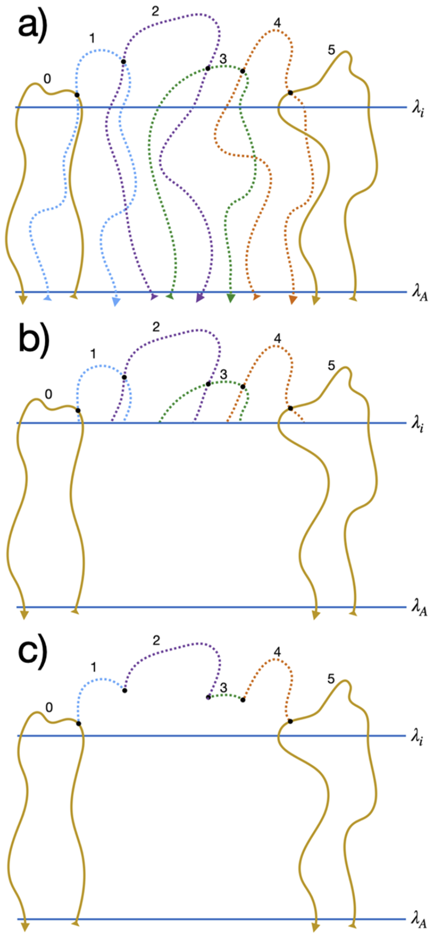


FIG. 2. Illustration of wasted MD steps in shooting and WF. (a) shows six consecutive paths being generated by the shooting move where only the solid golden paths, with index 0 and 5, are being saved. (b) gives an equivalent scenario in the WF algorithm showing that considerably fewer MD steps are needed to obtain the same paths 0 and 5 via N_s = 5 subtrajectories. Still, WF is not as thrifty as SS and WT since only parts of the subtrajectories, shown in (c), actually contribute to the sampling progress to get from path 0–5. The additional steps in (b) are seemingly "wasted" but still needed for the superdetailed balance relation. SS and WT do not generate wasted MD steps but rely on a one-step crossing condition as discussed in the main text.

illustrates a hypothetical MC sequence in path sampling of six consecutive paths, labeled 0–5, where the shooting point has an order parameter larger than λ_i. If only every fifth path is saved, only paths 0 and 5 are considered as in Fig. 2(a). Although the intermediate paths contribute for their decorrelation, it is clear that many MD steps can be omitted, as exploited by the subtrajectory moves. Figure 2(b) shows a scenario where the same final path is being generated with a set of hypothetical WF subtrajectories resembling the top scenario. Instead of five full trajectories, only four short subtrajectories and one full trajectory are needed to establish a new full path (path 5) from the old one (path 0). Based on this principle alone, the relative efficiency gain η of subtrajectory moves compared to standard shooting is expected to be

$$\eta(N_s) = \frac{N_s L_p}{L_p + (N_s - 1)L_s}, \tag{24}$$

where L_p and L_s are, respectively, the average length of a full path and a subpath. Still, if we purely focus on the MD steps that are required to allow for the progression from path 0 to path 5, even fewer MD steps are needed, as shown in Fig. 2(c). Yet, the "extra" (wasted) MD steps in panel (b) are required for the superdetailed balance, as discussed in Sec. III. Wasted MD steps are avoided in SS and WT where the shooting always happens at an interface [see Figs. 1(a) and 1(b)]. The price to be paid for this is the additional complication with regard to the one-step crossing condition (see Sec. VI). However, even with a slightly higher MD waste, the WF move requires considerably fewer MD steps than standard shooting.

Equation (24) levels off to a constant L_p/L_s for increasing N_s. Likewise, Eqs. (22) and (23) show that the trend $\mathcal{N}(N_s) = \mathcal{N}(1)/N_s$ is not sustained for increasing N_s as $\mathcal{N}$ ultimately levels off to 1. It is henceforth assumed that while efficiency initially increases quite rapidly as a function of N_s, it cannot surpass L_p/L_s and ultimately even decreases when $\mathcal{N}(N_s)$ levels off. Clearly, for the $[0^-]$ and $[0^+]$ ensemble where $L_p = L_s$, no gain is expected, and one could set $N_s = 1$ if data storage latency would not be an issue. Therefore, as a rule of thumb, N_s can be set approximately equal to L_p/L_s such that for $L_p > L_s$, the cost of the MC move is less than doubled, while Eq. (24) reaches more than 50% of its anyways unattainable maximum of L_p/L_s.

Although the essence of the above analysis is correct, there is, however, a caveat: rejections leave a much heavier mark on the subtrajectory move than on standard shooting. If, for instance, the extension of the fifth and last subpath in Fig. 2(b) is rejected, it would imply a complete reset to the latest accepted full path (path 0) since subpath 4 is not a valid trajectory and extending subpath 4 after the rejection would violate detailed balance. As a result, all MD steps of subpaths 1–5 are trashed as the next move starts from path 0 again. Instead, the MC chain will only fall back to path 4 (assuming path 4 was accepted) in standard shooting. It is therefore clear that rejections in the subtrajectory move approach should be avoided even more than in the shooting method. This can be achieved with the high-acceptance procedure that is discussed in Sec. V.

V. HIGH-ACCEPTANCE PROCEDURE

As discussed in Sec. IV, a rejection in the subtrajectory moves implies a large amount of wasted MD steps. An early rejection scheme, such as the one used in TIS and RETIS with standard shooting (see Sec. III), is also not so helpful as a rejection cannot be made until the generation of the last subtrajectory has been initiated. It is, therefore, preferable to combine the subtrajectory moves with the *high-acceptance* scheme.[17] The approach uses the following two tricks. First, if the final subtrajectory has a backward extension ending in state B, the MC move is not directly rejected. Instead, the extension forward in time is completed, and if it ends in state A, the path is time-reversed, providing an $A \to B$ path. The consequence is that the time-direction selection probability $P_{\text{sel}}(\text{td})$ in Eq. (4) is no longer 0.5 for all paths as an $A \to B$ path can be generated in two ways: either by choosing the correct time-direction immediately or in reverse. This implies an extra factor two in the generation probabilities P_{gen}, in Eqs. (1) and (10), of the $A \to B$ paths compared to $A \to A$ paths. We henceforth write

$$\frac{P_{\text{gen}}(\text{path}^{(n)} \to \text{path}^{(o)} \text{ via } \bar{\chi})}{P_{\text{gen}}(\text{path}^{(o)} \to \text{path}^{(n)} \text{ via } \chi)} = \frac{P(\text{path}^{(o)})q(\text{path}^{(o)})/M^{(n)}}{P(\text{path}^{(n)})q(\text{path}^{(n)})/M^{(o)}}, \tag{25}$$

where

$$q(\text{path}) = \begin{cases} 1 & \text{if path} \in \{A \to A\}, \\ 2 & \text{if path} \in \{A \to B\}. \end{cases} \tag{26}$$

The second trick is to slightly change the sampling distribution. Instead of sampling the correct physical path distribution, $P(\text{path})$, restrained to the path ensemble's requirements, an alternative path distribution $\tilde{P}(\text{path})$ is sampled. From Eqs. (1) and (25), the acceptance probability thus becomes

$$P_{\text{acc}} = \min\left[1, \frac{\tilde{P}(\text{path}^{(n)})P(\text{path}^{(o)})q(\text{path}^{(o)})M^{(o)}}{\tilde{P}(\text{path}^{(o)})P(\text{path}^{(n)})q(\text{path}^{(n)})M^{(n)}}\right], \tag{27}$$

and to maximize the acceptance, we choose the sampling distribution in ensemble $[i^+]$ as

$$\tilde{P}(\text{path}) = P(\text{path})w_i(\text{path})\mathbf{1}_{[i^+]}(\text{path})$$

with

$$w_i(\text{path}) = q(\text{path})M_{\lambda_i}(\text{path}), \tag{28}$$

where $\mathbf{1}_C(x)$ is the indicator function that equals 1 if x is part of set C and 0 otherwise. A subscript λ_i was added to the last term M, as the number of equal probable possibilities for a first shooting, generally depends on the interface λ_i. Substituting Eq. (28) in Eq. (27) implies that with high-acceptance,

$$P_{\text{acc}} = \mathbf{1}_{[i^+]}(\text{path}^{(n)}). \tag{29}$$

In other words, the new path will always be accepted unless the MC move led to a path not obeying the ensemble's definition: starting at λ_A, ending at λ_A or λ_B, and having at least one crossing with λ_i. By construction, the crossing of λ_i is always achieved in the subtrajectory moves if the starting condition at λ_A is met. Hence, the only necessary rejection is when the extension of the final successful subtrajectory ends at λ_B in both time-directions.

If no successful subtrajectories were generated after N_s attempts, s^0 could be extended. However, since this would regenerate the old trajectory in deterministic dynamics and otherwise a trajectory that is highly correlated with the old one, it is preferable to reject the move. Other potential reasons for rejections could be due to non-convergence of the atomistic forces in AIMD level calculations. Another potential issue is jumpy order parameters[42] such that M_{λ_i} can be zero even if the path is actually valid. This issue is further discussed in Sec. VII.

Exact natural averages can still be obtained by weighting each sample j with the inverse of $w_i(j)$. For instance, the estimated local crossing probability, previously defined by Eq. (13), can now be expressed as

$$p(m) = \frac{\sum_{j=0}^{m-1} w_i(j)^{-1} h_j}{\sum_{j=0}^{m-1} w_i(j)^{-1}} \approx P_A(\lambda_{i+1}|\lambda_i). \tag{30}$$

The effect of the weighting implies that different samples have different contribution. If a sample j' has a much lower than average w_i^{-1} factor, the sample could essentially be removed from Eq. (30) without significantly affecting the estimate $p(m)$. Yet, thanks to this sample not being rejected, sample $j' + 1$ is more different than $j' - 1$ than it would be in the case that j' was rejected. This shows the power of the high-acceptance approach.

The improved acceptance in the subtrajectory move will slightly reduce the acceptance in the replica exchange move. For instance, if a path j from ensemble $[i^+]$ will be exchanged with a path k from ensemble $[(i+1)^+]$, the acceptance becomes[17]

$$P_{\text{acc}} = \mathbf{1}_{[(i+1)^+]}(j) \times \min\left[1, \frac{w_i(k) w_{(i+1)}(j)}{w_i(j) w_{(i+1)}(k)}\right]. \tag{31}$$

Without high-acceptance, the factor in Eq. (31) after the multiplication sign equals 1. This means that whenever j, the path originating from $[i^+]$ is valid for $[(i+1)^+]$, the swap will be accepted. Note that any path in $[(i+1)^+]$ is also valid in $[i^+]$. This lower acceptance is not dramatic since replica exchange moves do not require any MD steps. Therefore, replica exchange moves have negligible CPU cost. The only exception is the $[0^-] \leftrightarrow [0^+]$ swap in which two new paths are generated. Without high-acceptance, this move is always accepted. For SS and WT, the acceptance remains 100%, but this is not the case for WF. We can solve this problem for WF in RETIS by sampling the $[0^-]$ and $[0^+]$ ensembles with the standard shooting method without high acceptance. Due to this w_{0^+} and w_{0^-} equal 1 irrespective to the paths and swapping between these two ensembles will always be accepted. The absence of high-acceptance is partly compensated by early rejection (see Sec. III). Moreover, in these ensembles, there is no difference between the average path length of a subpath and a full path, making the subtrajectory moves anyways not so effective for these ensembles.

The high-acceptance protocol eliminates the more serious drawbacks of rejections in the subtrajectory moves compared to shooting. In Sec. VI, we discuss how the one-step crossing condition can be met.

VI. ONE-STEP CROSSING CONDITION

As discussed above, SS and WT are very thrifty algorithms with respect to the number of generated MD steps. Yet, the one-step crossing condition puts a challenge to the implementation. One can eliminate the one-step crossing condition via the new but less thrifty WF algorithm that is further discussed in Sec. VII. In this section, we discuss a few algorithmic solutions to overcome the one-step crossing condition in SS and WT. These two approaches assume that one time step in (RE)TIS is effectively also one MD step.

The one-step crossing can be achieved in different ways. The most straightforward way is to generate velocities from a Maxwell–Boltzmann distribution, execute an MD step, and calculate the new order parameter, and if the crossing is established, then the two frames comprising the crossing are extended at the side above λ_i to create a new subpath. The problem with this approach is that after each velocity generation, an MD step, and therefore a force calculation, is required. In particular, if λ_i is at a steep slope of the potential energy surface, the two trajectory frames forming the crossing of a given interface might be rather far apart in λ-space. In such cases, if one of the two frames is located in the very proximity of the interface, it might be extremely unlikely to re-generate a new one-step crossing from the configuration furthest to the λ_i interface, given a random approach to generate velocities.

There are essentially two strategies to reduce the cost for fulfilling the one-step criterion: (i) generate atom velocities from a Maxwell–Boltzmann distribution and predict the next step's order parameter without performing an actual MD step and (ii) generate velocities in a way such that the crossing is likely achieved after very few attempts. Strategy (i) assumes that generating new velocities is rather computationally inexpensive, and the expense of the one-step crossing condition is mostly provided by the force calculation. This is the case for AIMD level simulations as these typically consist of just a few (hundreds of) atoms while requiring a high CPU demand for the force calculation. In large classical MD systems with a significant number of atoms, the velocity generation might actually be equally expensive as a force calculation. In that case, strategy (ii) might be preferable.

A. Prediction strategy

The velocity-Verlet[29] MD integrator propagates a phase point $x(t) = (r(t), v(t))$ deterministically to a next phase point $x(t + \Delta t) = (r(t + \Delta t), v(t + \Delta t))$. The integrator is most conveniently expressed via "intermediate velocities" at $t + \Delta t/2$,

$$\begin{aligned} v(t+\Delta t/2) &= v(t) + f(t)\Delta t/(2m), \\ r(t+\Delta t) &= r(t) + v(t+\Delta t/2)\Delta t, \\ v(t+\Delta t) &= v(t+\Delta t/2) + f(t+\Delta t)\Delta t/(2m), \end{aligned} \tag{32}$$

where m is the mass and f are the forces. We used a simplified notation here, but one should realize that for an N particle system, both r, v, and f are 3N-dimensional vectors and m is actual a $3N \times 3N$ diagonal mass matrix. Furthermore, the forces are determined from the positions: $f(t) = f(r(t))$.

Equation (32) suggests that one MD step requires two force evaluations, but this is not the case when the steps of Eq. (32) are called repeatedly in a loop. After the force calculation at the third step, required to determine $v(t + \Delta t)$, the forces are stored such that these can be used at the first step of the next cycle. With the same reasoning, if the forces are known already at time t from its previous step, a new force evaluation is only needed to determine $v(t + \Delta t)$, but not $r(t + \Delta t)$. This means that if the order parameter only depends on geometry, $\lambda = \lambda(r)$, its value at $t + \Delta t$ can also be determined without the need of doing an actual force calculation.

When testing the one-step crossing for the selected configuration with randomized velocities, a new (single step) MD trajectory is started with no information available from the previous MD step. However, the selected configuration is also part of the previous subpath, so the corresponding forces could have been known, in principle. When not available, the forces can be reobtained from the trajectory data without further electronic structure calculation in AIMD or from the gradient of force field potential in classical MD. In particular, let $x_1 = (r_1, v_1)$ and $x_2 = (r_2, v_2)$ be two consecutive phase points of the latest subpath that define a crossing. This means that x_2 follows from x_1 through a single MD step and both points are at opposite sides of the interface. Therefore, both points are viable points for shooting off the next subpath. By inverting Eq. (32), we can derive

$$f_1 = \frac{2\,m\,(r_2 - r_1 - \Delta t\,v_1)}{\Delta t^2}, \quad f_2 = \frac{2\,m\,(r_1 - r_2 + \Delta t\,v_2)}{\Delta t^2}. \tag{33}$$

Hence, Eq. (33) directly provides the forces on the two potential shooting points by reading the trajectory data from the subpath. Given that one of these two points is selected as a shooting point and new randomized velocities are generated, the coordinates after one MD step can be determined without any additional force calculation but using just the first two steps of Eq. (32). Hence, the value of the order parameter after one step can be asserted.

If the prediction suggests that a crossing might be achieved, the MD step is completed and then the next subtrajectory is generated. If the velocities do not lead to a crossing, a new velocity randomization is attempted until the crossing condition is met. As in SS, the shooting point selection has to be maintained, and the computation of Eq. (33) only needs to be done once for the generation of each subpath. Naturally, if the MD step integrator is more complex than velocity-Verlet (due to thermostats, barostats, constraints, and stochasticity), then the prediction becomes more difficult. The method also works best if a MD step is computationally expensive while regenerating velocities is relatively cheap. This method is therefore more suitable for simulations with the AIMD level. The approach has been implemented in the PyRETIS software, and it can be directly used with the CP2K[25] external MD engine. Note that the use of the plain velocity-Verlet MD integrator is rather common in path sampling since the generation of paths is already thermostated via the shooting move that allows a change of energy, while the individual paths have NVE dynamics.

B. Alternative velocity generation

The mathematically simple form of Eq. (10) is due to the many terms conveniently canceling out. For instance, the terms in Eq. (9), $\rho_r(r^{0,3})$, $\rho_r(r^{4,3})$, and $\rho_r(r^{4,6})$ in $P_{\text{gen}}(\text{path}^{(o)} \to \text{path}^{(n)} \text{ via } \chi)$ cancel out with, respectively, $\rho_r(r^{3,0})$, $\rho_r(r^{4,3})$, and $\rho_r(r^{6,4})$ in $P_{\text{gen}}(\text{path}^{(n)} \to \text{path}^{(o)} \text{ via } \overline{\chi})$ because $r^{\alpha,\beta} = r^{\beta,\alpha}$. However, whereas consecutive (accepted) subtrajectories share a common configuration point, they do not necessarily share of common phase point as $v^{\alpha,\beta} \neq v^{\beta,\alpha}$. Here, $v^{\alpha,\beta}$ refers to the velocities of s^β at the configuration point $r^{\alpha,\beta}$, and $v^{\beta,\alpha}$ refers to the velocities of s^α at an identical configuration point. These velocities have typically not the same orientation nor amplitude. Luckily, the $\rho_v(v^{\alpha,\beta})$ terms still cancel out via $P_{\text{gen}}(v^{\alpha,\beta}) = \rho_v(v^{\alpha,\beta})$ and $\rho(x^{\alpha,\beta}) = \rho_r(r^{\alpha,\beta})\rho_v(v^{\alpha,\beta})$ in Eq. (8).

Now, suppose that in a N particle system not all $3N$ velocity components are regenerated from a Maxwell–Boltzmann distribution, but some velocities components are kept and some others are inverted (multiplied with -1). These two velocity groups do not cancel out in Eq. (8) as they are not part of P_{gen}, which implies that the final results changes from $P(s^6)/\rho_r(r^{4,6})$ to $P(s^6)/[\rho_r(r^{4,6})\rho_v(u^{4,6})]$, where $u^{4,6}$ are the velocity components that are either unchanged or inverted. Since the equilibrium velocity distribution is symmetric $\rho_v(v) = \rho_v(-v)$ and $u^{4,6}$ is identical to $u^{6,4}$ except for some components having different sign, all the $\rho_v(u^{\alpha,\beta})$ terms cancel in the ratio, Eq. (10), just like the $\rho_r(r^{\alpha,\beta})$ terms.

This allows for different strategies. For instance, if the dynamics is stochastic, all velocities can simply be inverted. This option was used for WT in Ref. 17. Inverting the velocities of specific atoms or molecules whose coordinates determine the order parameter could also be effective. The other velocities could be either kept unchanged, randomized, or a combination. For instance, in protein folding, simulations inverting the velocities of all protein atoms while leaving the velocities of the solvent molecules (partly) unchanged would make the sampling less diffusive. Reinspection of Eq. (32) shows that the coordinates of the atoms with the inverted velocities are mapped exactly back after 1 MD step to the previous coordinates regardless of the velocities of the other atoms. As a result, the one-step crossing condition is automatically fulfilled.

This approach requires, however, a single MD step resolution at the interface crossing. In large molecular systems, it is not desirable to save trajectory coordinates every MD step as it could overwhelm hard disk capacity and will result in a loss of effective CPU efficiency due to the time that is spent writing to disk. An adaptive scheme could be adopted when the frequency of order parameter determination and the data retention is intensified whenever the system approaches an interface. Since trajectories can later be swapped in a replica exchange move, this adaptive approach would have to be carried out for all interfaces or, at least, in the proximity of neighboring interfaces. The latter choice might still lead to path ensembles receiving a trajectory missing the right resolution at the relevant interface. That part of the trajectory would then have to be reintegrated by MD. While all these issues can be solved in theory, it puts quite some challenges to the implementation. Moreover, if the integrator is not deterministic but involves a thermostat or barostat, the one-step crossing might still not be guaranteed. Several velocity generation steps might still be needed. These challenges lead us to derive the WF move that straightforwardly can be implemented in present path sampling codes, such as OpenPathSampling[18,19] and PyRETIS,[20,21] with, potentially, any MD engine.

VII. WIRE FENCING

Compared to the SS and WT moves, the shooting point selection of the WF move is constructed to avoid the one-step crossing issue altogether. Instead of restricting the shooting point to sets of crossing points at an interface, WF allows any phase point between the path ensemble's specific ensemble interface, λ_i, and interface λ_B to be picked. To increase the efficiency of the WF move in systems with asymmetric free energy barriers (see Fig. 3), the selection range and the boundaries of the subtrajectories can be changed by replacing $\lambda_{\rm B}$ with a user-defined *cap*-interface, $\lambda_{\rm cap}$ with $\lambda_i < \lambda_{\rm cap} \le \lambda_{\rm B}$ value.

The presence of a relatively flat downhill region after the barrier's maximum and before a stable product state implies that transition paths can become very long. If accepted, the paths will have a large fraction of points at the right side of the free energy barrier from which shooting has a very high chance to generate a failed $\lambda_B \to \lambda_B$ trajectory. This problem was also addressed by the spring-shooting method.[43]

In an AIMD level simulation of aqueous silicate condensation,[44] this issue was solved by defining λ_B in the RETIS algorithm at the position of $\lambda_{\rm cap}$ in the figure. After the simulation was completed, all paths reaching λ_B were extended in a straightforward MD simulation. The introduction of the $\lambda_{\rm cap}$ interface makes these post-simulation MD extensions redundant.

We will first outline the WF algorithm without a cap-interface (or $\lambda_{\rm cap} = \lambda_B$) using the high-acceptance protocol. The introduction of $\lambda_{\rm cap}$ only requires a few modifications that we discuss afterward.

1. From the old path, count the number of frames $M^{(o)}_{\lambda_i}$ between λ_i and λ_B. If $M^{(o)}_{\lambda_i} = 0$, we immediately reject the full MC move. Otherwise, continue with the next step.
2. Subdivide the $M^{(o)}_{\lambda_i}$ points into groups where each group are the points lying on a segment connecting λ_i with λ_B or a segment connecting λ_i with itself.
3. Select one segment as s^0 based on a weighted random selection such that each segment has a chance to be selected proportional to the number of points it has.
4. Set two counters n_s and n_a equal to zero: $n_s = n_a = 0$. Then, start the following loop: steps 5–12.
5. Select at random one of the configuration points of the last subpath, s^{n_s}, as the new shooting point.
6. Generate random velocities from a Maxwell–Boltzmann distribution.
7. Starting from the configuration point with the new velocities, apply the MD integrator to go backward and forward in time until λ_i or $\lambda_{\rm B}$ is crossed.
8. Increase the n_s counter by one: $n_s = n_s + 1$.
9. If both time-directions crosses $\lambda_{\rm B}$, the trial subpath is rejected. In that case, the previous successful subpath is kept, $s^{n_s} = s^{n_s-1}$. Go to step 12. Otherwise, continue with the next step.
10. Increase the n_a counter by one: $n_a = n_a + 1$.
11. Accept the trial subpath such that it becomes s^{n_s}.
12. If $n_s < N_s$, go to step 5. Otherwise, continue with next step.
13. If no accepted subpaths have been generated, $n_a = 0$, stop and reject the move. Otherwise, continue with the next step.
14. Extend the last subpath s^{N_s} in both time-directions with MD until $\lambda_{\rm A}$ or $\lambda_{\rm B}$ is hit. If the path ends at $\lambda_{\rm B}$ at both time-directions, the whole MC move is rejected. Otherwise, continue to the next step.
15. If the path is $\lambda_B \to \lambda_A$, reverse the time-direction of the path.
16. Now, a new full path has successfully been established. Let $q^{(n)}$ be 2 if it is a $\lambda_A \to \lambda_B$ path. Otherwise, it is 1. Let $M^{(n)}_{\lambda_i}$ be the number of frames between λ_i and λ_B. The weight-factor of the path is $w^{(n)} = q^{(n)} M^{(n)}_{\lambda_i}$ that is needed for computing proper path ensemble averages, Eq. (30), and for a possible swap move via Eq. (31).

The scenario of the potential rejection at step 1 is shown in Fig. 4(a), which can occur due to a jumpy character of the order parameter.[42] A typical example is nucleation where the time steps in path sampling is usually chosen to consist of many MD steps[45] for the reason that computing order parameters for nucleation is rather costly. As a result, occasionally the order parameter, defined by the size of the largest cluster, can make sudden jumps such that more than one interface is crossed in a single RETIS time step.

The path shown in Fig. 4(a) is a valid path in $[i^+]$ such that $\mathbf{1}_{[i^+]} = 1$, but $w_i = 0$ since $M_{\lambda_i} = 0$. In a WF move, such a path has zero probability to be generated. Yet, its contribution in Eq. (30) to the average, if hypothetically sampled, would be $w_i^{-1} = \infty$, and therefore, the sampling average becomes ill-defined. This can be solved by not allowing $w = 0$ weights,

$$w_i(\text{path}) = \min[1, q(\text{path}) M_{\lambda_i}(\text{path})]. \tag{34}$$

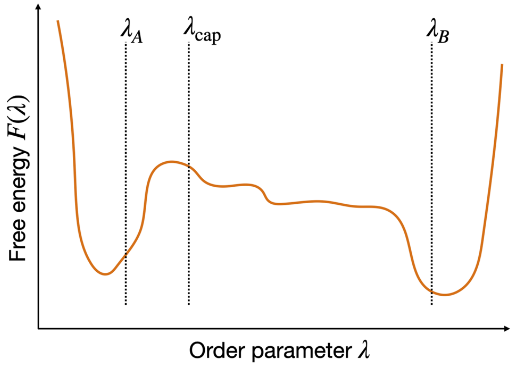


FIG. 3. Illustration of an asymmetric barrier where the placement of a cap-interface, $\lambda_{\rm cap}$, in WF can avoid the generation of long subtrajectories and too many shooting points being in the basin of attraction of state *B*.

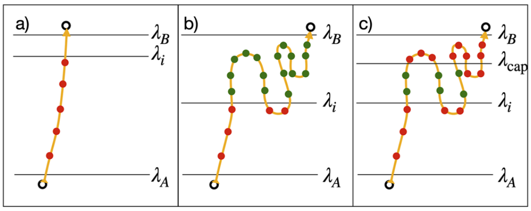


FIG. 4. Illustration of the s^0 selection from the old path. Selectable shooting points are shown in green, end-points by open black circles, and all other points in red. (a) shows the "jumpy order parameter" case that leads to an immediate rejection as no selectable points are present. (b) and (c) show the selectable points without and with cap-interface, respectively.

Introducing this small modification of Eq. (28) solves the "division by zero" problem and has further no impact of the implementation nor on the robustness of the algorithm. The existence of jumpy trajectories implies that a pure WF simulation is no longer ergodic. A path like the one in Fig. 4(a) can never be made from a WF move, and vice versa, it cannot be destroyed by the WF move if it is fed as the initial path to the algorithm. However, the full sampling remains ergodic due to the replica exchange moves.

Step 2 is further illustrated in Fig. 4(b). We can identify two groups of selectable shooting points (in green), one group of seven points lying on a $\lambda_i \to \lambda_i$ segment and one group of nine points on a $\lambda_i \to \lambda_B$ segment. Hence, these segments are selected as s^0 with a 7/16 and 9/16 probability, respectively. In the next step, the points of the selected segment have an equal probability to be selected for the first shooting.

Despite that all the green points have the same 1/16 probability to be selected for shooting off the first subpath, the two-step selection process is needed to fix s^0. With a single step selection, it could be possible to first obtain a failed trial path t^1 that starts from a point at the first group, followed by a successful subtrajectory that is launched from a point of the second group. This will break the superdetailed balance as it would not be possible to generate t^1 from s^0 in the reverse path [see the example construction paths in Eqs. (2) and (3)].

The introduction of the cap-interface changes the initial s^0 selection, as is shown in Fig. 4(c), where, for the same path as panel (b), there are now three groups of two points that can be chosen. Note that not all the points between λ_i and λ_{cap} are selectable as the points on a $\lambda_{\text{cap}} \to \lambda_{\text{cap}}$ segment should be excluded. The algorithm is further identical as described above with λ_{cap} instead of λ_B in the main loop (steps 5–12). Outside the main loop (step 13–15), λ_B is not replaced by λ_{cap} since the final extension always shall reach the A or B states. In the final step (16), $M_{\lambda_i}^{(n)}$ is replaced with the number of frames between λ_i and λ_{cap} excluding those on $\lambda_{\text{cap}} \to \lambda_{\text{cap}}$ segments.

VIII. NUMERICAL RESULTS

We tested the WF algorithm on three model systems: a simple one-dimensional system for which we can perform full RETIS simulations with high convergence and two challenging complex systems based on classical MD and AIMD, where our analysis is more qualitative based on a single path ensemble simulation. The one-dimensional system describes a single particle in a double-well potential that is moving following the underdamped Langevin equation as previously described in Ref. 33. The purpose of these simulations is to show numerically that the WF method leads indeed to exact results. In addition, due to the high degree of convergence that can be reached, we also draw some conclusions on the efficiency compared to standard shooting. However, it should be taken into account that a larger boost factor is expected for more complex high-dimensional systems.

The other two systems are part of ongoing projects on which we plan to report extensively in later publications. The classical MD system describes the thin film breakage in oil–water mixtures based on the studies Refs. 46–49. The system size of this simulation is over 100 000 atoms, making the one-step crossing impracticable as it requires a stop/restart at every MD step. Instead, in our single path ensemble simulation, the coordinates were recorded every 50 MD steps. The AIMD system describes the electron transfer between ruthenium ions in a redox reaction taking place in liquid water. To determine the relative position of the moving electron, the Kohn–Sham orbitals are projected on maximally localized Wannier Functions[50] whose centers can be viewed as "electron positions." This implies that in order to compute the order parameter from a configuration point, a full electronic structure calculation is required. A cheap prediction scheme as described in Sec. VI is therefore not suitable. For both systems, we show the usefulness of the cap-interface in practical simulations.

A. Double-well 1 D barrier

Despite the model's simplicity, several popular rare event simulation methods, such as forward flux sampling (FFS)[51,52] and other splitting based methods,[53–55] have shown that they can easily fall into a kind of sampling trap when applied to this system, yielding a too low rate and non-time-symmetric transition paths.[33]

The double-well barrier system consists of a one-dimensional particle moving in the following potential:[33]

$$V(z) = z^4 - 2z^2, \tag{35}$$

with underdamped Langevin dynamics. In reduced units, the Boltzmann constant and mass are set to unity, $k_B = m = 1$, while the temperature and friction coefficient are set equal to $T = 0.07$ and $\gamma = 0.3$. The equations of motion are propagated using an MD time step equal to $\mathrm{d}t = 0.025$. In a straightforward MD run, the particle will mostly oscillate within one of the potential minima at $z = -1$ and $z = 1$ but also (very) infrequently cross the transition state at z = 0. During the oscillatory movement, the total energy of the particle will fluctuate by the random force of the Langevin dynamics. As a result, the system is effectively two-dimensional in phase space where the velocity can be considered as an orthogonal degree of freedom. The reason that FFS and other splitting type methods underestimate the crossing rate is due to an insufficient sampling of the tail in the velocity distribution.[33] Path sampling methods, such as RETIS, which are based on both forward and backward in time propagation, do not have this issue.

We defined eight RETIS interfaces: $\lambda_A = \lambda_0 = -0.99$, $\lambda_1 = -0.8$, $\lambda_2 = -0.7$, $\lambda_3 = -0.6$, $\lambda_4 = -0.5$, $\lambda_5 = -0.4$, $\lambda_6 = -0.3$, and $\lambda_B = \lambda_7 = 1.0$ and ran four RETIS simulations using the PyRETIS code[20,21] consisting of 200 000 cycles. In all simulations (Shooting, WF, WF*, and WF-cap), each path ensemble either employs only shooting or only WF as the main MC move in addition to replica exchange moves. In the simulation "Shooting," all path ensembles employ the shooting move. In the other simulations, the WF move is used for most path ensembles. However, simulation WF* uses normal shooting in the $[0^-]$ ensemble, while simulations WF and WF-cap use the shooting move in both the $[0^-]$ and $[0^+]$ ensemble as was suggested in Sec. III. The WF-cap simulation uses a cap-interface at $\lambda_{\text{cap}} = 0.1$. At each cycle, all path ensembles are updated with an ensemble move (shooting or WF) or with replica exchange moves with a 50%–50% probability. In case that a replica exchange move is selected, another 50%–50% probability determines whether the $[0^-] \leftrightarrow [0^+]$, $[1^+] \leftrightarrow [2^+]$, $\ldots$, $[5^+] \leftrightarrow [6^+]$ swaps will be attempted or the $[0^+] \leftrightarrow [1^+]$, $[2^+] \leftrightarrow [3^+], \ldots, [4^+] \leftrightarrow [5^+]$ swaps. In the latter case, the $[0^-]$ and

$[6^+]$ ensembles simply duplicate the previous path (null move). In the WF simulations, the number of subpaths was arbitrarily set equal to $N_s = 6$ for all path ensembles.

The results are shown in Fig. 5 and in Table I where they are compared with Kramers' theory,[56] which, for this system, can be considered as a nearly exact reference. Figure 5 shows that the WF based simulations rapidly converge close to the Kramers' value of the rate, confirming the exactness of the superdetailed balance relations and the correct implementation in the PyRETIS code. The results based on shooting are further off but have a significantly lower computational cost per RETIS cycle (see Table I).

Based on the relative errors from the block averaging analysis and the cost per cycle, we can compute the CPU efficiency time for each method, shown in the last column. Based on these numbers, we can see that the WF, WF* and WF-cap simulations are 2.5, 2.4, and 2.7 times more efficient than the simulation in which all path ensembles use the standard shooting move as their main MC move. Note that an improvement of more than a factor 2 is rather remarkable, given the low dimensionality of the system.

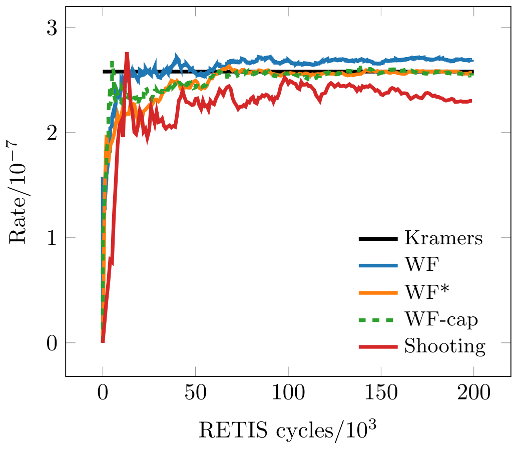


FIG. 5. Total running average of the computed rate as function of RETIS cycles.

In Table II, we further examine the acceptance probabilities of the different moves. It is apparent that in all simulations, the main MC move has a nearly 100% acceptance in the path ensembles where the WF move is employed, thanks to the high-acceptance protocol. The acceptance is marginally lower at the last path ensembles $[5^+]$ and $[6^+]$ from which there is a higher probability to generate $\lambda_B \to \lambda_B$ paths. The shooting move has a lower acceptance but has the advantage that all swapping moves with the $[0^-]$ ensemble are accepted if shooting is the main move in both $[0^-]$ and $[0^+]$. Since $[0^-]$ can only swap with $[0^+]$, these are the computationally expensive $[0^-] \leftrightarrow [0^+]$ swaps.

The other swapping moves are inexpensive as they do not require any MD steps. Therefore, an anticipated lower acceptance for these swapping moves in the WF simulations would not be dramatic. However, even this is not always the case. At first sight, this appears counter-intuitive. Given a pair of paths in two neighboring ensembles, the standard swap should always have an acceptance probability that is equal to or higher than the acceptance based on Eq. (31). However, this effect can be canceled by the path distributions not being the same. Since the altered path distribution in the high-acceptance scheme, Eq. (28), overrepresents paths with many points between λ_i and λ_B or λ_{cap}, the $[i^+]$ path ensemble is likely to contain a higher fraction of paths crossing λ_{i+1}. From the data of Table II, this seems indeed the case in the majority of path ensembles.

B. Thin film breakage

A system of 1100 dodecane molecules layered on a slab of 23 936 water molecules is studied in the NPT ensemble via full atom TIS simulations using the GROMACS 2020.1 simulation package[22] as the external engine. The dodecane molecules are simulated according to the OPLS-AA force field[57] and the water molecules with the TIP4p/2005 model.[58] The preparation of the initial equilibrated system is explained in detail by Ref. 46. The temperature is set to 300 K and is controlled with a velocity rescaling method,[59] employing a coupling time of 0.1 ps. Pressure is controlled by the Berendsen barostat, and its normal component is maintained constant at 1 bar,

TABLE I. Simulation data for the double-well 1D barrier system. The cost column describes the total number of calculated MD steps. The errors are based on block averaging using single standard deviations. The final column shows the CPU efficiency times[37] corresponding to the number of required MD steps for obtaining a relative error equal to 1. Simulation "Shooting" uses the standard shooting move as the main MC move in all path ensembles. The other simulations use the WF move in all ensembles except for $[0^-]$ (WF, WF*, and WF-cap) and $[0^+]$ (WF and WF-cap). WF-cap uses a cap-interface at $\lambda_{\text{cap}} = 0.1$.

Simulation	Rate/10^{-7}	ϵ_r (%)	Cost/10^7	Cost$\cdot\epsilon_r^2$/10^{11}
Shooting	2.30	6.46	5.32	222.0
WF	2.69	2.28	16.98	88.3
WF*	2.58	2.19	19.56	93.9
WF-cap	2.54	2.29	15.72	82.4
Kramers	2.58			

TABLE II. Acceptance ratio (%). Simulation "Shooting" uses the standard shooting move as main MC move in all path ensembles. The other simulations use the WF move in all ensembles except for $[0^-]$ (WF, WF*, and WF-cap) and $[0^+]$ (WF and WF-cap).

	Shooting		WF		WF*		WF-cap	
Ens.	Main	Swap	Main	Swap	Main	Swap	Main	Swap
$[0^-]$	84.6	100.0	84.3	100.0	84.5	83.9	84.3	100.0
$[0^+]$	84.2	57.8	84.0	55.6	100.0	49.0	84.0	55.8
$[1^+]$	48.8	15.5	100.0	16.3	100.0	17.9	100.0	16.8
$[2^+]$	37.8	13.4	100.0	19.9	100.0	19.7	100.0	20.2
$[3^+]$	32.2	11.5	100.0	18.5	100.0	18.0	100.0	18.4
$[4^+]$	30.1	12.2	100.0	20.6	100.0	20.3	100.0	20.4
$[5^+]$	30.0	14.7	99.8	28.2	99.9	28.3	100.0	26.6
$[6^+]$	29.1	16.7	99.2	33.9	99.2	34.2	100.0	30.8

with a time constant of 1.0 ps and a compressibility coefficient of $4.7 \cdot 10^{-5}$ bar^{-1}. The velocity-Verlet algorithm is used to solve the Newton equations of motion with a timestep of 0.002 ps. Periodic boundary conditions are applied in all directions, with the z direction being perpendicular to the 2D film. The box size is set to equal a box size of $15 \times 15 \times 5.1983$ nm^3.

The order parameter of the system is calculated by discretizing the system into 85×85 tiles in the x and y direction such that the order parameter value becomes the number of empty dodecane tiles that also have empty neighbors in the x and y direction. Such a definition provides a way to measure the presence of low-density regions, in addition to any breakage or "hole" formation that occurs within a trajectory. The sensitivity of the order parameter is determined by the specified discretizing size. In our case, the order parameter values fluctuated between 0 and 5 during an equilibrium run at $T = 300$ K. Based on this, we set $\lambda_A = 5$. We further defined $\lambda_B = 100$ as preliminary analysis showed that from this point on the hole tends to grow further with a negligible chance to close again.

To obtain an initial reactive trajectory, we ran an equilibrium run at $T = 375$ K until the thin film broke down. For our single path ensemble analysis, we further defined $\lambda_\mathrm{i} = 10.0$ as the interface that has to be crossed. In addition, we set the cap-interface $\lambda_\mathrm{cap} = 15.0$. We then created 1000 trajectories using standard shooting and WF with $N_s = 10$. Three exemplary trajectories from the WF simulation are shown in Fig. 6(a).

From the sample size of 1000 MC moves, the acceptance in WF was equal to 73.4% and 35.0% for standard shooting. The limited sample size prohibits accurate CPU efficiency analysis, but a qualitative assertion of the sampling effectivity can be obtained by viewing the simulated path lengths as function of the MC step.

Figure 7(a) shows that the WF sampling has much more frequent transitions between long and short paths, whereas shooting is mostly stuck in the short path domain. Once the shooting move manages to produce a long path, the path remains in the MC chain due to a long series of rejections (e.g., around step 500 where the same path length remains for a number of steps due to rejections). This indicates that the shooting move is struggling to properly sample path space. Even if the acceptance is not extremely low for the short paths, it fails to make regular switches to the longer paths. Moreover, if a long path is generated, the subsequent moves are likely rejected such that other longer paths are not likely found.

C. Ruthenium–ruthenium self-exchange reaction

We studied the self-exchange reaction between two ruthenium ions in aqueous solution described by the following chemical reaction:

$$Ru^{2+} + Ru^{3+} \rightarrow Ru^{3+} + Ru^{2+}. \quad (36)$$

The simulation system consisted of two ruthenium ions, 63 H_2O molecules and one OH^- ion. The dynamics were propagated using NVE velocity-Verlet and the CP2K[25] simulation package. The effect of temperature was introduced via the randomization of velocities from a Maxwell–Boltzmann distribution at a temperature of 300 K. We used a time step of 0.5 fs, and periodic boundary conditions were applied to a cubic box with an edge length of 12.4138 Å.

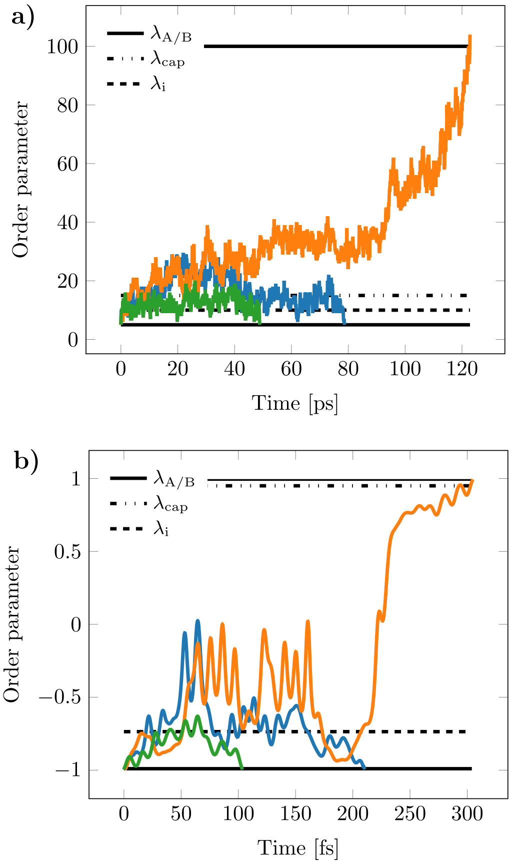


FIG. 6. Exemplary trajectories from the WF algorithm in the $[i^+]$ path ensemble showing the progress of the order parameter vs time. The stable state interfaces λ_A, λ_B, the cap-interface λ_{cap}, and the ensemble interface λ_i are shown as well. The two different panels represent the (a) classical MD level simulation of the thin film breakage and (b) the AIMD level simulations of the ruthenium self-exchange reaction.

Further simulation details on functional and basis sets are explained in Ref. 60.

To monitor the reaction progress, the electron transfer has been "followed" by transforming the occupied Kohn–Sham orbitals[61] into maximally localized Wannier functions (MLWFs)[50] and computing the distance between the center of these localized functions (X) describing the moving electron to each of the ruthenium ions. The order parameter of the system is then defined as

$$\lambda = \frac{(d_{\mathrm{Ru}-X} - d_{\mathrm{Ru}'-X})}{d_{\mathrm{Ru}-\mathrm{Ru}'}}, \quad (37)$$

where $d_{\mathrm{Ru}-X}$ is the distance between X and the initial ruthenium electron donor, $d_{\mathrm{Ru}'-X}$ is the distance between X and the initial

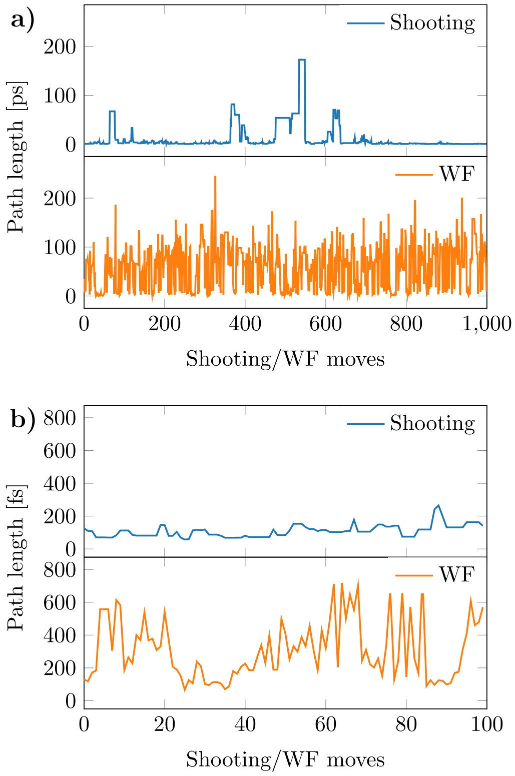


FIG. 7. Path length vs MC move for WF and standard shooting for (a) classical MD system of thin film breakage and (b) AIMD system of the ruthenium self-exchange reaction.

ruthenium electron acceptor, and $d_{\mathrm{Ru-Ru'}}$ is the distance between the two ruthenium ions in the system. In this formulation, $\lambda = -1$ and $\lambda = +1$ define the reactant state and product state, respectively. $\mathrm{Ru}^{2+}/\mathrm{Ru}^{3+}$ have 5/6 d-electrons and H_2O/OH^- have eight valence electrons. This means there are a total of 523 MLWFs in the system. The order parameter, Eq. (37), requires the location X of the transferring electron, which is one of the centers of these 523 MLWFs. To identify which is X, each Wannier center is linked to either a ruthenium or oxygen atom that is closest. Then, if one ruthenium ion has six associated MLWFs, X is set to be the one that is the farthest away from this ruthenium ion. In the case that both ruthenium ions have five associated MLWFs, one of the oxygens has an excess MLWF (9 instead of 8), and the center that is farthest away from this oxygen is set as X.

To qualitatively compare standard shooting and WF for this system, we run two single path ensemble simulations representing $[i^+]$ with $\lambda_i = -0.736$, $\lambda_A = -0.99$, and $\lambda_B = +0.99$. The value for $\lambda_i = -0.736$ was chosen from preliminary runs where we aimed for a 20% probability that a path ends up at state B. In the WF simulation, an additional $\lambda_{\mathrm{cap}} = +0.95$ was set to avoid $\lambda_B \to \lambda_B$ rejections due to the selection of shooting points lying within the basin of attraction of state B. Here, we only applied a rather modest number of subtrajectories $N_s = 2$. Higher performances might be obtained with a larger number of subpaths. Exemplary trajectories of the WF simulation are shown in Fig. 6(b).

Due to the relatively low value of N_s, the subpath contribution to the total WF computational cost is only 15%. The acceptance probability increased from the shooting move's 48% to WF's 96%. Similar to the classical MD system, the WF simulation seems to show a better sample exploration when we look at the path length as function of the MC step [Fig. 7(b)]. The standard shooting algorithm seems not to be able to produce any paths larger than 300 fs. The WF algorithm, however, started with a short initial path but was able to quickly move up to the 600 fs range and making regular transitions between the shorter and longer paths. Hence, also here, the sampling quality of the WF algorithm appears substantially superior to the one of standard shooting.

IX. CONCLUDING REMARKS

We reviewed the recently developed subtrajectory moves stone skipping (SS) and web throwing (WT) and added a new member to this group: wire fencing (WF). These moves are more efficient than the standard shooting move, which has been the main MC move for path sampling simulations during the last two decades. The subtrajectory moves proceed from a complete old path to a complete new path via a series of intermediate short paths (subpaths/subtrajectories). While this increases the average cost of a MC step, the correlations between paths are substantially reduced leading to a lower statistical inefficiency. The use of shorter paths resembles approximate path sampling methods, such as PPTIS or milestoning. However, the subtrajectory moves are still exact like standard shooting as they are based on mathematically rigorous superdetailed balance relations. The approach is preferably combined with a high-acceptance protocol in which the sampling distribution of the paths is adjusted in order to maximize the acceptance of newly generated trajectories. The effect of the biased distribution is undone in the post-simulation analysis using appropriate reweighting. The SS and WT move, however, require a one-step crossing condition, which complicates their implementation, and we discussed several solutions for this issue. The new WF does not rely on the one-step crossing condition and is, therefore, the most practical solution to the aforementioned problem even if it is slightly more wasteful than SS and WT. The WF move is, in particular, useful when the path sampling code uses an external MD engine and/or when the computation of the order parameter is costly. We showed the exactness and the efficiency gain of the WF approach in a RETIS simulation where the transition rate of an underdamped Langevin particle in a double-well potential has been computed and compared with the analytical Kramers' expression. Thereafter, we showed qualitatively how the WF move performs in a classical MD system, describing the thin film breaking process, and in an AIMD level system, describing an electron transfer process between ruthenium ions in aqueous solution. In both cases, the WF move seems to allow a faster sampling through path space than standard shooting, which was concluded from the rapid switches that WF made between the shorter and longer paths.

ACKNOWLEDGMENTS

We acknowledge funding from the Research Council of Norway (toppforsk project Theolight, Grant No. 275506) and computational resources from NOTUR (Project No. nn9254k).

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Daniel T. Zhang: Conceptualization (equal); Formal analysis (equal); Investigation (lead); Methodology (equal); Software (equal); Visualization (equal); Writing – original draft (equal); Writing – review & editing (equal). **Enrico Riccardi**: Conceptualization (equal); Methodology (equal); Software (equal); Supervision (equal); Writing – review & editing (equal). **Titus S. van Erp**: Conceptualization (equal); Formal analysis (equal); Methodology (equal); Supervision (equal); Visualization (equal); Writing – original draft (equal); Writing – review & editing (equal).

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request. The algorithmic developments have been implemented in the current pyretis2.dev version (the current release is PyRETIS-2[21]) and will be included in the forthcoming main release (PyRETIS-3). The code and input files for the double-well 1D barrier system is already available at https://gitlab.com/pyretis following the FAIR principle for scientific data and software and data.[62,63]

APPENDIX: ANALYTICAL EXPRESSIONS FOR THE STATISTICAL INEFFICIENCY IN MODEL SYSTEMS

Section IV introduces a model where at each MC move j, there is a chance of π_R that the state of the system remains essentially unchanged and a chance of $\pi_M = 1 - \pi_R$ to "throw a dice." The latter implies that at step j, the output value of h_j equals 1 with a probability p and 0 with a probability $1 - p$. Let us consider the conditional probability that $h_j = 0$, given that $h_0 = 0$: $P(h_j = 0|h_0 = 0)$. We can distinguish two scenarios. Scenario 1 relates to the case that all j moves implied a "remain," and therefore, $h_j = h_0 = 0$. Scenario 2 is related to the situation that at least once the dice was thrown. In this scenario, we have that h_j is either 1 or 0 with respective probabilities p and $1 - p$. The probability of having scenario 1 equals π_R^j and that of scenario 2 equals $1 - \pi_R^j$. Therefore,

$$\begin{aligned} P(h_j = 0|h_0 = 0) &= \pi_R^j + (1 - \pi_R^j)(1 - p) \\ &= (1 - p) + p\pi_R^j. \end{aligned} \tag{A1}$$

Likewise, we can derive all the other conditional probabilities,

$$\begin{aligned} P(h_j = 1|h_0 = 0) &= (1 - \pi_R^j)p = p - p\pi_R^j, \\ P(h_j = 0|h_0 = 1) &= (1 - \pi_R^j)(1 - p) = (1 - p) + (1 - p)\pi_R^j, \\ P(h_j = 1|h_0 = 1) &= \pi_R^j + (1 - \pi_R^j)p = p + (1 - p)\pi_R^j. \end{aligned} \tag{A2}$$

Let us call $p_{kl} = P(h_j = k \wedge h_0 = l) = P(h_j = k|h_0 = l)P(h_0 = l)$. From Eqs. (A1) and (A2), we can derive

$$\begin{aligned} p_{00} &= (1 - p)^2 + p(1 - p)\pi_R^j, \\ p_{10} &= p(1 - p) - p(1 - p)\pi_R^j = p_{01}, \\ p_{11} &= p^2 + p(1 - p)\pi_R^j, \end{aligned} \tag{A3}$$

and from this, we can compute

$$\begin{aligned} \langle (h_0 - p)(h_j - p) \rangle = p_{00}p^2 - p_{10}(1 - p)p \\ - p_{01}(1 - p)p + p_{11}(1 - p)^2. \end{aligned} \tag{A4}$$

In the above expression, all the π_R-independent terms cancel. This is expected since we know the result equals 0 if $\pi_R = 0$. The remaining π_R-dependent terms sum up to

$$\begin{aligned} &p(1 - p)\pi_R^j[p^2 + 2p(1 - p) + (1 - p)^2] \\ &\quad = p(1 - p)\pi_R^j[p + (1 - p)]^2 = p(1 - p)\pi_R^j. \end{aligned} \tag{A5}$$

From Eqs. (16), (17), and (A5), we derive that

$$C(j) = \pi_R^j \Rightarrow n_c = \frac{\pi_R}{1 - \pi_R}, \tag{A6}$$

and via Eq. (15),

$$\mathcal{N} = 1 + 2\frac{\pi_R}{1 - \pi_R} = \frac{1 + \pi_R}{1 - \pi_R}. \tag{A7}$$

As $\pi_M = 1 - \pi_R$, (A7) is equivalent to Eq. (19) of Sec. IV.

In the second model, we assume $\pi_R = 0$, but there are two phases $x = \alpha, \beta$ that have, respectively, probabilities P_α and P_β and local crossing probabilities p_α and p_β. Let $\pi_\rho = 1 - \pi_\mu$ be the probability that the MC maintains the previous phase. The inverse probability π_μ implies throwing the dice to determine the phase x such that the selection probability for x corresponds to P_α and $P_\beta = 1 - P_\alpha$. After the phase x is set, h_j will be set to 1 or 0 with respective probabilities p_x and $(1 - p_x)$. Given that the phase of the first sample equals $x_0 = x$, the chance that the jth sample has the same or opposite phase equals, respectively, $\pi_\rho^j + (1 - \pi_\rho^j)P_x$ and $(1 - \pi_\rho^j)(1 - P_x)$. This leads to the following conditional probabilities akin Eqs. (A1) and (A2),

$$\begin{aligned} P(h_j = 0|x_0 = x) &= \pi_\rho^j(1 - p_x) + (1 - \pi_\rho^j)(1 - p) \\ &= \pi_\rho^j(p - p_x) + (1 - p) \\ &= \pi_\rho^j P_y(p_y - p_x) + (1 - p), \\ P(h_j = 1|x_0 = x) &= \pi_\rho^j p_x + (1 - \pi_\rho^j)p \\ &= \pi_\rho^j(p_x - p) + p \\ &= \pi_\rho^j P_y(p_x - p_y) + p, \end{aligned} \tag{A8}$$

where $y \in (\alpha, \beta)$ and $y \neq x$. Hence, analogous to Eq. (A3),

$$\begin{aligned} p_{k0} &= \sum_{x=\alpha,\beta} P_x(1 - p_x)P(h_j = k|x_0 = x), \\ p_{k1} &= \sum_{x=\alpha,\beta} P_x p_x P(h_j = k|x_0 = x), \end{aligned} \tag{A9}$$

which leads to

$$\begin{aligned}
p_{00} &= (1-p)^2 + \pi_\rho^j \sum_x P_x P_y (1-p_x)(p_y-p_x) \\
&= (1-p)^2 + \pi_\rho^j P_\alpha P_\beta (p_\alpha - p_\beta)^2, \\
p_{10} &= p(1-p) + \pi_\rho^j \sum_x P_x P_y (1-p_x)(p_x-p_y) \\
&= p(1-p) - \pi_\rho^j P_\alpha P_\beta (p_\alpha - p_\beta)^2 = p_{01}, \\
p_{11} &= p^2 + \pi_\rho^j \sum_x P_x P_y p_x (p_x-p_y) \\
&= p^2 + \pi_\rho^j P_\alpha P_\beta (p_\alpha - p_\beta)^2. \qquad \text{(A10)}
\end{aligned}$$

Analogous to Eqs. (A4) and (A5), we find that

$$\langle (h_0 - p)(h_j - p) \rangle = \pi_\rho^j P_\alpha P_\beta (p_\alpha - p_\beta)^2, \qquad \text{(A11)}$$

and like Eq. (A6),

$$\begin{aligned}
C(j) &= \frac{P_\alpha P_\beta (p_\alpha - p_\beta)^2}{p(1-p)} \pi_\rho^j = K_s \pi_\rho^j \\
\Rightarrow n_c &= K_s \frac{\pi_\rho}{1-\pi_\rho} = K_s \frac{1-\pi_\mu}{\pi_\mu}, \qquad \text{(A12)}
\end{aligned}$$

where we used $\pi_\mu = 1 - \pi_\rho$ and Eq. (21). Substitution of Eq. (A12) in Eq. (15) leads to Eq. (20).

REFERENCES

[1] K. Lindorff-Larsen, S. Piana, R. O. Dror, and D. E. Shaw, Science **334**, 517 (2011).

[2] D. E. Shaw, P. J. Adams, A. Azaria, J. A. Bank, B. Batson, A. Bell, M. Bergdorf, J. Bhatt, J. A. Butts, T. Correia, R. M. Dirks, R. O. Dror, M. P. Eastwood, B. Edwards, A. Even, P. Feldmann, M. Fenn, C. H. Fenton, A. Forte, J. Gagliardo, G. Gill, M. Gorlatova, B. Greskamp, J. Grossman, J. Gullingsrud, A. Harper, W. Hasenplaugh, M. Heily, B. C. Heshmat, J. Hunt, D. J. Ierardi, L. Iserovich, B. L. Jackson, N. P. Johnson, M. M. Kirk, J. L. Klepeis, J. S. Kuskin, K. M. Mackenzie, R. J. Mader, R. McGowen, A. McLaughlin, M. A. Moraes, M. H. Nasr, L. J. Nociolo, L. O'Donnell, A. Parker, J. L. Peticolas, G. Pocina, C. Predescu, T. Quan, J. K. Salmon, C. Schwink, K. S. Shim, N. Siddique, J. Spengler, T. Szalay, R. Tabladillo, R. Tartler, A. G. Taube, M. Theobald, B. Towles, W. Vick, S. C. Wang, M. Wazlowski, M. J. Weingarten, J. M. Williams, and K. A. Yuh, in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC '21* (Association for Computing Machinery, New York, NY, USA, 2021).

[3] M. E. Goldberg, G. V. Semisotnov, B. Friguet, K. Kuwajima, O. B. Ptitsyn, and S. Sugai, FEBS Lett. **263**, 51 (1990).

[4] B. Peters, *Reaction Rate Theory and Rare Events* (Elsevier, Amsterdam, The Netherlands, 2017).

[5] T. S. van Erp, D. Moroni, and P. G. Bolhuis, J. Chem. Phys. **118**, 7762 (2003).

[6] T. S. van Erp, Phys. Rev. Lett. **98**, 268301 (2007).

[7] C. Dellago, P. G. Bolhuis, F. S. Csajka, and D. Chandler, J. Chem. Phys. **108**, 1964 (1998).

[8] R. Cabriolu, K. M. S. Refsnes, P. G. Bolhuis, and T. S. van Erp, J. Chem. Phys. **147**, 152722 (2017).

[9] A. Arjun and P. G. Bolhuis, J. Phys. Chem. B **124**, 8099 (2020).

[10] M. Moqadam, A. Lervik, E. Riccardi, V. Venkatraman, B. K. Alsberg, and T. S. van Erp, Proc. Natl. Acad. Sci. U. S. A. **115**, E4569 (2018).

[11] M. Eigen and L. de Maeyer, Proc. R. Soc. London, Ser. A **247**, 505 (1958).

[12] W. C. Natzle and C. B. Moore, J. Phys. Chem. **89**, 2605 (1985).

[13] D. Moroni, P. G. Bolhuis, and T. S. van Erp, J. Chem. Phys. **120**, 4055 (2004).

[14] A. K. Faradjian and R. Elber, J. Chem. Phys. **120**, 10880 (2004).

[15] S. Roet, D. T. Zhang, and T. S. van Erp, J. Phys. Chem. A **126**, 8878 (2022).

[16] C. Dellago, P. G. Bolhuis, and D. Chandler, J. Chem. Phys. **108**, 9236 (1998).

[17] E. Riccardi, O. Dahlen, and T. S. van Erp, J. Phys. Chem. Lett. **8**, 4456 (2017).

[18] D. W. H. Swenson, J.-H. Prinz, F. Noe, J. D. Chodera, and P. G. Bolhuis, J. Chem. Theory Comput. **15**, 813 (2019).

[19] D. W. H. Swenson, J.-H. Prinz, F. Noe, J. D. Chodera, and P. G. Bolhuis, J. Chem. Theory Comput. **15**, 837 (2019).

[20] A. Lervik, E. Riccardi, and T. S. van Erp, J. Comput. Chem. **38**, 2439 (2017).

[21] E. Riccardi, A. Lervik, S. Roet, O. Aarøen, and T. S. van Erp, J. Comput. Chem. **41**, 370 (2020).

[22] E. Lindahl, M. J. Abraham, B. Hess, and D. van der Spoel,GROMACS 2020.1 Manual. https://doi.org/10.5281/ZENODO.3685920 (2020).

[23] S. Plimpton, J. Comput. Phys. **117**, 1 (1995).

[24] P. Eastman, M. S. Friedrichs, J. D. Chodera, R. J. Radmer, C. M. Bruns, J. P. Ku, K. A. Beauchamp, T. J. Lane, L.-P. Wang, D. Shukla, T. Tye, M. Houston, T. Stich, C. Klein, M. R. Shirts, and V. S. Pande, J. Chem. Theory Comput. **9**, 461 (2013).

[25] J. Hutter, M. Iannuzzi, F. Schiffmann, and J. VandeVondele, Wiley Interdiscip. Rev.: Comput. Mol. Sci. **4**, 15 (2014).

[26] A. Stukowski, Modell. Simul. Mater. Sci. Eng. **20**, 045021 (2012).

[27] S. Winczewski, J. Dziedzic, and J. Rybicki, Comput. Phys. Commun. **198**, 128 (2016).

[28] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, J. Chem. Phys. **21**, 1087 (1953).

[29] D. Frenkel and B. Smit, *Understanding Molecular Simulations from Algorithms to Applications* (Academic Press, San Diego, CA, 2002).

[30] J. I. Siepmann and D. Frenkel, Mol. Phys. **75**, 59 (1992).

[31] T. J. H. Vlugt, R. Krishna, and B. Smit, J. Phys. Chem. B **103**, 1102 (1999).

[32] W. K. Hastings, Biometrika **57**, 97 (1970).

[33] T. S. van Erp, Adv. Chem. Phys. **151**, 27 (2012).

[34] A. Ghysels, S. Roet, S. Davoudi, and T. S. van Erp, Phys. Rev. Res. **3**, 033068 (2021).

[35] C. Dellago and P. G. Bolhuis, Mol. Simul. **30**, 795 (2004).

[36] T. S. van Erp and P. G. Bolhuis, J. Comput. Phys. **205**, 157 (2005).

[37] T. S. van Erp, J. Chem. Phys. **125**, 174106 (2006).

[38] A. M. Ferrenberg and R. H. Swendsen, Phys. Rev. Lett. **63**, 1195 (1989).

[39] T. S. van Erp, M. Moqadam, E. Riccardi, and A. Lervik, J. Chem. Theory Comput. **12**, 5398 (2016).

[40] J. Rogal, W. Lechner, J. Juraszek, B. Ensing, and P. G. Bolhuis, J. Comput. Phys. **133**, 174109 (2010).

[41] E. Vanden-Eijnden, M. Venturoli, G. Ciccotti, and R. Elber, J. Comput. Phys. **129**, 174102 (2008).

[42] A. Haji-Akbari, J. Chem. Phys. **149**, 072303 (2018).

[43] Z. F. Brotzakis and P. G. Bolhuis, J. Chem. Phys. **145**, 164112 (2016).

[44] M. Moqadam, E. Riccardi, T. T. Trinh, A. Lervik, and T. S. van Erp, Phys. Chem. Chem. Phys. **19**, 13361 (2017).

[45] D. Moroni, P. R. ten Wolde, and P. G. Bolhuis, Phys. Rev. Lett. **94**, 235703 (2005).

[46] O. Aarøen, E. Riccardi, T. S. v. Erp, and M. Sletmoen, Colloids Surf., A **632**, 127808 (2022).

[47] O. Aarøen, E. Riccardi, and M. Sletmoen, RSC Adv. **11**, 8730 (2021).

[48] E. Riccardi and T. Tichelkamp, Colloids Surf., A **573**, 246 (2019).

[49] E. Riccardi, K. Kovalchuk, A. Y. Mehandzhiyski, and B. A. Grimes, J. Dispersion Sci. Technol. **35**, 1018 (2014).

[50] N. Marzari, A. A. Mostofi, J. R. Yates, I. Souza, and D. Vanderbilt, Rev. Mod. Phys. **84**, 1419 (2012).

[51] R. J. Allen, C. Valeriani, and P. R. ten Wolde, J. Phys.: Condes. Matter **21**, 463102 (2009).

[52] F. A. Escobedo, E. E. Borrero, and J. C. Araque, J. Phys.: Condes. Matter **21**, 333101 (2009).

[53] T. E. Booth and J. S. Hendricks, Nucl. Technol./Fusion **5**, 90 (1984).

[54] P. G. Melnik-Melnikov and E. S. Dekhtyaruk, Probab. Eng. Mech. **15**, 125 (2000).

[55] M. Villenaltamirano and J. Villenaltamirano, in *Queueing, Performance and Control in ATM*, North-Holland Studies in Telecommunication, edited by J. W.

Cohenand C. D. Pack (Elsevier Science Publisher B. V., Amsterdam, 1991), Vol. 15, pp. 71–76, 13th International Teletraffic Congress (ITC-13), Copenhagen, Denmark, Jun 19-26, 1991.

[56] P. Hänggi, P. Talkner, and M. Borkovec, Rev. Mod. Phys. **62**, 251 (1990).

[57] W. L. Jorgensen, D. S. Maxwell, and J. Tirado-Rives, J. Am. Chem. Soc. **118**, 11225 (1996).

[58] J. L. F. Abascal and C. Vega, J. Chem. Phys. **123**, 234505 (2005).

[59] G. Bussi, D. Donadio, and M. Parrinello, J. Chem. Phys. **126**, 014101 (2007).

[60] A. Tiwari and B. Ensing, Faraday Discuss. **195**, 291 (2016).

[61] W. Kohn and L. J. Sham, Phys. Rev. **140**, A1133 (1965).

[62] E. Riccardi, S. Pantano, and R. Potestio, Interfaces: Focus **9**, 20190005 (2019).

[63] A.-L. Lamprecht, L. Garcia, M. Kuzak, C. Martinez, R. Arcila, E. Martin Del Pico, V. Dominguez Del Angel, S. Van De Sandt, J. Ison, P. A. Martinez *et al.*, Data Sci. **3**, 37 (2020).

PAPER C

PNAS

RESEARCH ARTICLE | CHEMISTRY
BIOPHYSICS AND COMPUTATIONAL BIOLOGY

Highly parallelizable path sampling with minimal rejections using asynchronous replica exchange and infinite swaps

Daniel T. Zhang[a,1], Lukas Baldauf[a,1], Sander Roet[b], Anders Lervik[a], and Titus S. van Erp[a,2]

Edited by Pablo Debenedetti, Princeton University, Princeton, NJ; received October 26, 2023; accepted January 9, 2024

Capturing rare yet pivotal events poses a significant challenge for molecular simulations. Path sampling provides a unique approach to tackle this issue without altering the potential energy landscape or dynamics, enabling recovery of both thermodynamic and kinetic information. However, despite its exponential acceleration compared to standard molecular dynamics, generating numerous trajectories can still require a long time. By harnessing our recent algorithmic innovations—particularly subtrajectory moves with high acceptance, coupled with asynchronous replica exchange featuring infinite swaps—we establish a highly parallelizable and rapidly converging path sampling protocol, compatible with diverse high-performance computing architectures. We demonstrate our approach on the liquid–vapor phase transition in superheated water, the unfolding of the chignolin protein, and water dissociation. The latter, performed at the ab initio level, achieves comparable statistical accuracy within days, in contrast to a previous study requiring over a year.

rare events | path sampling | asynchronous replica exchange | infinite swapping | Markov-chain Monte Carlo

The capacity to rapidly and accurately simulate molecular transition phenomena holds the potential to significantly enhance chemical discoveries, thereby advancing catalytic processes (1), optimizing drug molecule design (2), and guiding self-assembly for various applications, such as organic photovoltaics (3). However, dynamic processes like chemical reactions, nucleation, or protein (un)folding usually hinge on rare molecular events, rendering direct molecular dynamics (MD) simulations ineffective (4). A way to bridge the time gap is to use rare event sampling techniques like the Markov chain Monte Carlo (MC)-based transition path sampling (TPS) method, which involves the collection of numerous short MD trajectories (5).

Transition interface sampling (TIS) (6) and, even more efficiently, replica exchange TIS (RETIS) (7) build upon this idea to calculate quantitative dynamical properties through a series of path sampling simulations, each targeting a distinct path ensemble reflecting different stages of the transition. Each trajectory evolves on the true potential energy surface, and the sampling of trajectories follows the same distributions as what would result if the relevant trajectories were extracted from a hypothetically long MD run. Yet, the distinctive feature of path sampling simulations lies in their computational emphasis on actual barrier-crossing events, which stands in contrast to plain MD where the computational effort is primarily directed toward explorations within stable states. Despite exponential speedup compared to direct MD, the study of complex systems can still require months of simulation time due to the necessity of generating numerous trajectories for achieving the required statistical precision.

In this paper, we leverage recent algorithmic innovations that achieve such results in a matter of days or weeks. This transformative progress is driven by harnessing four recent algorithmic innovations delineated in refs. 8 and 9. Initially, we improve the core MC path generation move, opting for a sequence of intermediate short subtrajectories, yielding enhanced decorrelation from the preceding trajectory upon acceptance. Subsequently, by slightly adjusting the sampling distribution and compensating through reweighting, we maximize the acceptance. Third, the integration of an asynchronous replica exchange scheme facilitates seamless swapping between path ensemble simulations, tackling the challenge of RETIS parallelization attributed to varying central processing unit (CPU) costs in path-generating MC moves. Last, we amplify the impact of computationally efficient replica exchange moves through the embrace of the infinite swapping limit (10), all while circumventing the need for combinatorially explosive computations.

While the mathematical proofs establishing the exactness of these algorithms were published in refs. 8 and 9, this article demonstrates their first implementation and efficient management of numerous realistic molecular processes in parallel, leveraging

Significance

Current molecular dynamics simulations have the capability to faithfully describe chemical reactions, nucleation, and protein folding events, providing essential information to drive the progress of technologies like catalysis and drug discovery. Nevertheless, even with the fastest supercomputers, a significant portion of these phenomena remains beyond the reach of standard simulations due to waiting times that can exceed billions of CPU years. While enhanced sampling techniques like path sampling allow for exponentially faster study of these events, obtaining converged results in experimentally relevant systems still typically spans from months to years. Here, we apply four innovative algorithmic enhancements that transform the state-of-the-art path sampling algorithm, replica exchange transition interface sampling (RETIS), into ∞RETIS, allowing convergence in a matter of days.

Author contributions: A.L. and T.S.v.E designed research; D.T.Z. and L.B. performed research; D.T.Z., L.B., S.R., A.L., and T.S.v.E. analyzed data; and D.T.Z., L.B., S.R., A.L., and T.S.v.E. wrote the paper.

The authors declare no competing interest.

This article is a PNAS Direct Submission.

[1] D.T.Z. and L.B. contributed equally to this work.

[2] To whom correspondence may be addressed. Email: titus.van.erp@ntnu.no.

Published February 5, 2024.

both classical and ab initio dynamics on high-performance computing (HPC) systems. Notably, the article showcases a large-scale simulation utilizing 40 graphics processing units (GPUs) in parallel— overcoming the challenge of effectively utilizing multiple GPUs for parallel molecular simulations.

Results

Path Sampling and RETIS Path Ensembles. MC techniques are valuable in various fields like statistical physics, finance, and artificial intelligence, where the common goal is to sample states following specific probability distributions. Within molecular simulations, MC is generally used to sample configuration space following the Boltzmann distribution. By an original insight from Pratt (11), the concept emerged that MC sampling could be applied to target the sampling of dynamic trajectories. In this framework, a trajectory referred to as X is depicted as a discrete sequence of phase points, called time slices $X = [x_0, x_1, \dots x_L]$. Each time slice x_i encapsulates the coordinates and velocities of atoms at a specific time $t = i\Delta t$. Here, L denotes the trajectory's length, and Δt represents a small time step. The path probability distribution equals $\rho(X) \propto \rho(x_0)p(x_0 \to x_1)p(x_1 \to x_2)\dots p(x_{L-1} \to x_L)$, where $\rho(x_0)$ is the equilibrium (Boltzmann) distribution of the first phase point and $p(x \to y)$ is the probability that the system's dynamics produces y after a Δt time step from x.

Applying this within an MC algorithm does not yet yield advantages over MD. However, the approach enables focusing on specific path ensembles defined by initial and terminal conditions, and/or reaction progress through sampling from a truncated distribution $\rho_E(X) = \rho(X) \cdot \mathbf{1}_E(X)$, where $\mathbf{1}_E(X)$ equals 1 if the path adheres to the ensemble E conditions, and 0 otherwise. To obtain dynamical quantitative results such as rates, a series of overlapping path ensemble simulations is needed. The RETIS ensembles possess both initial and final conditions, as well as a minimum progress requirement, that is gauged through a series of nonintersecting interfaces: $\lambda_0, \lambda_1, \dots, \lambda_n$ (Fig. 1*A*). These interfaces are hypersurfaces within phase space, often characterized by an order parameter $\lambda(x)$ that assigns a progress value to the reaction. The ith interface corresponds to the collection of phase points $\{x|\lambda(x) = \lambda_i\}$. The first and last interfaces define the reactant state A, $\{x|\lambda(x) < \lambda_A = \lambda_0\}$, and the product state B, $\{x|\lambda(x) > \lambda_B = \lambda_n\}$, respectively. The RETIS path ensemble $[i^+]$ encompasses all paths that commence by crossing λ_A toward the barrier region and conclude by either re-entering A or entering B. Moreover, each path within the ensemble is required to cross λ_i. This implies that the value of L is not fixed but varies for each path and the average path length typically increases with i. Alongside the $[i^+]$ ensembles, there exists an additional $[0^-]$ ensemble that explores the internal realm of state A (Fig. 1*B*).

Subtrajectory Moves. The primary MC move for generating paths has been the shooting move (12). It evolves by modifying the velocities of a random time slice of the old path, which is then propagated forward and backward in time using the MD time step integrator. Fine-tuning the shooting move necessitates a delicate balance between maximizing decorrelation and maintaining a satisfactory acceptance. When the adjustment to the shooting point is minimal, the resulting path often closely resembles its predecessor. Although this enhances the chance of the trial path being valid for the considered path ensemble, the substantial correlation among sampled paths necessitates a large number of trajectories to achieve low statistical errors. In subtrajectory moves, the creation of complete trial trajectories involves preceding them with several intermediate short subtrajectories. This ensures that successive accepted full trajectories do not share any configuration points and exhibit a greater degree of distinctiveness compared to shooting. The subtrajectories are only part of the inner workings of the MC move and are not stored or used for statistical analysis.

The initial subtrajectory moves (13) encountered certain implementation challenges which led to the development of the more flexible wire-fencing (WF) (8). The WF move involves a parameter N_s, representing the number of subtrajectories, and potentially a cap interface (Fig. 1*C*). Within the ensemble $[i^+]$, the WF move entails releasing a sequence of N_s short subtrajectories with termination criteria at λ_i or λ_{cap} (or λ_n if no cap is set). The first subtrajectory originates from a time slice of the previous full path with randomized Maxwell–Boltzmann velocities. Subsequent subtrajectories are generated from the previous successful subtrajectory until N_s attempts have been made. A subtrajectory ending at λ_{cap} in both time directions is classified as unsuccessful. After the completion of N_s subtrajectory trials, the last successful subtrajectory is integrated forward and backward in time until reaching either λ_0 or λ_n, resulting in the formation of a new full trajectory. Based on a Metropolis–Hastings acceptance/rejection scheme (14, 15), this new path can be accepted with a probability equal to

$$P_{\text{acc}} = \mathbf{1}_{[i^+]}(X^{(n)}) \times \min\left[1, \frac{M_{\lambda_i}(X^{(o)})}{M_{\lambda_i}(X^{(n)})}\right], \qquad [1]$$

where $X^{(o)}$ and $X^{(n)}$ are, respectively, the old and new paths, and $M_{\lambda_i}(X)$ is the number of possible shooting points for releasing a first subtrajectory from X. It was found that subtrajectory moves (with high acceptance) have the potential to improve RETIS' CPU efficiency by a factor of twelve compared to shooting (13).

High Acceptance. The high acceptance technique represents a significant enhancement for advanced shooting moves by introducing two algorithmic modifications: i) Reverse the time direction of the full trial path if it starts at λ_n and ends at λ_0 (Fig. 1*C*). ii) Modify the targeted sampling distribution from $\rho_{[i^+]}(X)$ to $\tilde{\rho}_{[i^+]}(X) = \rho_{[i^+]}(X) \cdot w_i(X)$, where $w_i(X)$ denotes the high-acceptance weight defined as:

$$w_i(X) = M_{\lambda_i}(X)q(X) \quad \text{with} \qquad [2]$$
$$q(X) = \begin{cases} 1 & \text{if } X \text{ is of the type } \lambda_0 \to \lambda_0 \\ 2 & \text{if } X \text{ is of the type } \lambda_0 \to \lambda_n \end{cases}.$$

These two actions culminate in the outcome that nearly all trajectories become viable for acceptance as P_{acc} of Eq. **1** becomes identical to the indicator function, $P_{\text{acc}} = \mathbf{1}_{[i^+]}(X^{(n)})$, which is always 1 for any trial trajectory within this scheme except for those ending at λ_n in both time directions. That chance is mostly negligible for all $[i^+]$ path ensembles except where λ_i is near the peak of the barrier or beyond. The only other reason for a rejection is when all N_s subtrajectories reach the cap-interface in both temporal directions. In that case, the "last successful" subtrajectory essentially comprises the path segment of the old path from which the initial shooting occurred. Extending this path segment likely generates a highly similar path to the old one, leading us to reject it in our WF implementation instead of investing CPU time in producing a significantly correlated path.

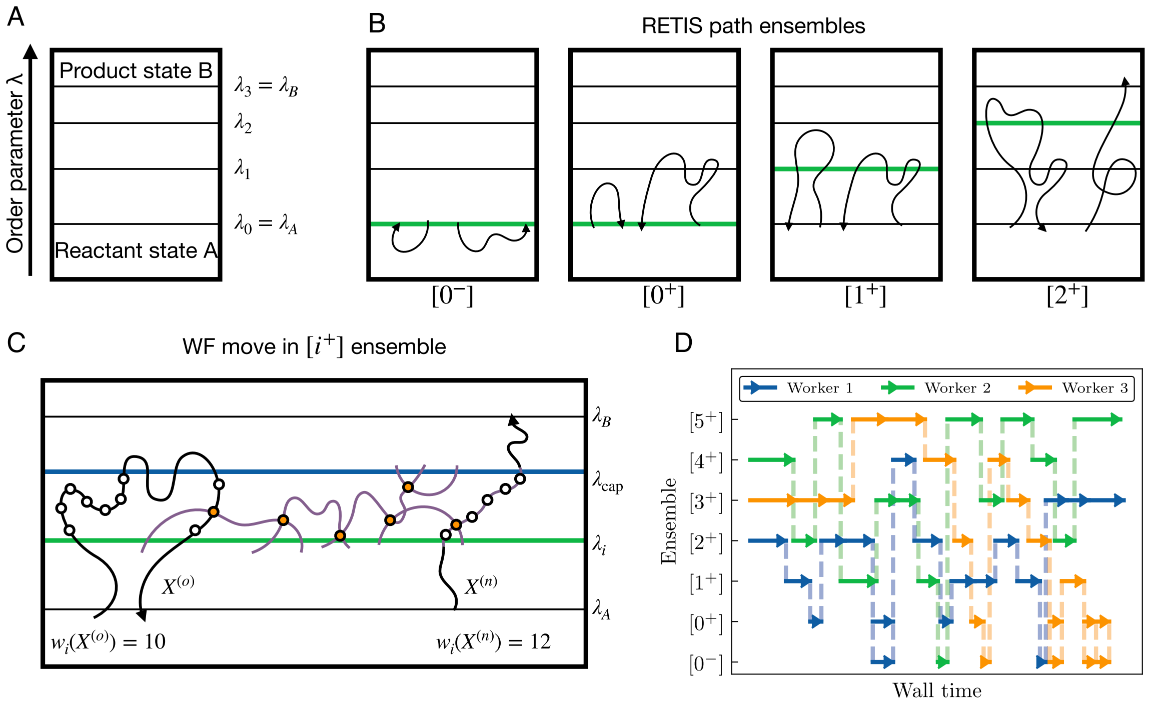


Fig. 1. RETIS path ensembles and path sampling methodology. (*A*) Concept of interfaces and states based on an order parameter λ (reaction coordinate). The horizontal axis represents an arbitrary additional order parameter. In this example, four interfaces are defined: λ_0, λ_1, λ_2, and λ_3. The first and last interfaces define the reactant and product states, respectively. (*B*) RETIS path ensembles. The minimal progress interface is highlighted in green. Two representative trajectories are shown for each ensemble. Two trajectories in $[0^+]$ and $[1^+]$ are identical, illustrating that ensembles overlap and paths may be sampled in several ensembles via a swapping move. (*C*) Demonstration of the WF move with $N_s = 6$. The fourth subtrajectory is unsuccessful and subsequently dismissed. Shooting points are indicated as orange circles and additional potential shooting points on both the old (X^o) and new (X^n) paths are represented by white circles. Shooting from a $\lambda_{\rm cap} \to \lambda_{\rm cap}$ segment is disallowed. With the extension of the last subtrajectory, a time direction is randomly chosen, which is flipped in the high-acceptance scheme if the resulting path is of the type $\lambda_B \to \lambda_A$. The resulting high-acceptance weights for the old and new paths are, respectively, 10 and 12, based on $M_{\lambda_i}(X^{(0)}) = 10$, $q(X^{(0)}) = 1$, $M_{\lambda_i}(X^{(n)}) = 6$, and $q(X^{(n)}) = 2$. (*D*) Time spent per ensemble per worker in an actual asynchronous replica exchange simulation double-well system (9). Arrows denote the moments when a worker completes a path to initiate the exchange of replicas between free ensembles. Minimal computational time is consumed during this process and concludes when the worker is randomly reassigned to another (or the same) free ensemble for a new path generation move. When both $[0^+]$ and $[0^-]$ ensembles are free, a point-exchange move is also incorporated into the random reassignment. In this move, the worker creates two new paths: one in $[0^+]$ by extending the endpoint of the $[0^-]$-path forward in time and another in $[0^-]$ by extending the starting point of the $[0^+]$-path backward in time (7).

The post-simulation analysis counteracts the impact of the distorted distribution by employing weighted averages for the sampled paths, assigning each path X a weight proportional to $1/w_i(X)$. In a simple one-dimensional double-well potential, the acceptance rate of the WF move stood at 100% for the path ensembles $[i^+]$ when λ_i was near state A and only slightly decreased to 99.2% in the ensemble closest to state B (8).

Asynchronous Replica Exchange. Despite TIS being significantly less efficient than RETIS, it has the advantage that its separate path ensembles can be simulated entirely autonomously, allowing parallel execution without communication overhead. In RETIS, however, path-generating MC moves within a single ensemble are alternatively succeeded by replica exchange moves between ensemble simulations. The irregular CPU costs of path-generating MC moves, stemming from the diverse path lengths, introduce synchronization challenges within a parallel RETIS simulation setup.

Assigning individual path ensembles with their own hardware setup leads to instances where hardware managing faster ensembles frequently remain idle, awaiting the completion of MC moves by their slower counterparts. For this reason, open-source path-sampling codes (16, 17) have implemented RETIS as a fully sequential algorithm. However, this design choice limits its potential to run simulations in a massively parallel manner.

In ref. 9, this challenge was addressed through an asynchronous replica exchange approach, where the number of path ensembles is set to be approximately double the number of hardware groups (referred to as "workers") that are assigned to execute path generation moves. This design ensures that, at any given moment, about half of the ensembles are "busily" engaged in path creation, while the other half is labeled as "free." Following the completion of an MC move by a worker, the ensemble it was assigned to and the newly formed path change status to free. Before the worker is randomly reassigned to one of the free ensembles for performing a new path generation move, a series of swapping moves take place between randomly selected pairs of free ensembles in which they attempt to exchange their current paths (Fig. 1*D*). For a selected ensemble pair, $[i^+]$ and $[j^+]$ with $j > i$ and, respectively, current paths X_i and X_j are swapped with an acceptance probability:

$$P_{\mathrm{acc}} = \mathbf{1}_{[j^+]}(X_i) \times \min\left[1, \frac{w_i(X_j)w_j(X_i)}{w_i(X_i)w_j(X_j)}\right], \qquad [3]$$

where term $\mathbf{1}_{[i^+]}(X_j)$ is omitted as $X_j \in [j^+]$ is always a valid trajectory for ensemble $[i^+]$ if $\lambda_j \geq \lambda_i$.

Ref. 9 demonstrated that asynchronous replica exchange significantly enhances wall time efficiency while minimally impacting CPU efficiency. Surprisingly, it even led to occasional improvements in CPU efficiency due to a more efficient distribution of CPU resources among different path ensembles. The algorithm tends to generate more trajectories in path ensembles with shorter average path lengths, which contributes positively to the overall efficiency.

Infinite Swapping. Generating a complete path may span minutes or hours, while evaluating Eq. 3 takes sub-seconds. This allows for numerous swap moves, but when does it become excessive? Plattner et al. (10) showed the feasibility of replicating the impact of executing an infinite number of swaps within finite CPU time, potentially maximizing the benefit of each swapping opportunity. To determine the frequency of sampling a specific state (path) in a particular ensemble after an infinite number of swaps, one only needs to sum over the probabilities of all permutations in which the considered state and ensemble are linked. While this method is efficient for a modest number of participating ensembles ($\lesssim$10), computational costs increase dramatically, transitioning from approximately a single second of wall time to millions of years as the number of ensembles grows from 7 to 20 (9).

Remarkably, this factorial scaling obstacle can be addressed by employing an expression based on weight matrices' permanents, which is equivalent to the summation of permutations (9). Despite being similar to the determinant, commonly taught in high school mathematics textbooks, the permanent is relatively unfamiliar among scientific researchers, potentially contributing to the lack of prior discovery of this relationship.

Like determinants, a matrix's permanent is recursively defined as the sum of permanents of reduced matrices with a row and column removed, but unlike determinants, it lacks alternating plus and minus signs. Recursive relations also involve factorial scaling, but faster methods exist for large matrices, such as Gaussian elimination for determinants, leading to third-order scaling.

Unfortunately, this technique does not extend to the computation of permanents, for which more complex approaches are necessary, characterized by steeper scaling (18, 19). Nonetheless, these methods are still considerably faster than factorial computations. Furthermore, since many elements of the weight matrices are zero, permanents only need to be computed for a limited number of low-dimensional sub-blocks of the weight matrix. As demonstrated in this paper, this enables us to conduct infinite swapping replica exchanges involving 80 ensembles and 40 workers, with the infinite swaps constituting only a minor portion of the CPU cost compared to that of path creation.

Application I: (Superheated) Water Boiling. We employ RETIS with the aforementioned algorithmic advancements, hereafter referred to as ∞RETIS (9), to study the liquid–vapor phase transition. Boiling phenomena have previously been explored using TPS (20, 21). However, the ∞RETIS approach offers a notable advantage in quantification, enabling the calculation of rates– a feat not easily achieved with the previous TPS method, even for the more common occurrence of surface boiling. Hence, the previous TPS studies were more qualitative than our current investigation and did not provide information on transition rates.

In the first boiling study, we aimed to compute the boiling rate in superheated water at 573.15 K. Superheated liquid is produced by gently heating a liquid beyond its boiling point (22). While establishing superheated water at this temperature is difficult experimentally, in our nano-sized simulation system devoid of nucleation sites like walls or impurities, the metastable liquid has a long lifetime. Consequently, this transition to vapor serves as a unique test for assessing our path sampling protocol's hardware scaling capabilities.

We conduct two simulations using different numbers of identical, GPU-equipped nodes. In the first simulation i), 20 interfaces and 10 workers run on one node, utilizing the NVIDIA Multi-Process Service (MPS) feature, which enables multiple compute unified device architecture (CUDA) processes or applications to share and utilize a single GPU. The other simulation ii) involves 40 interfaces and 20 workers, each operating on their exclusive nodes without MPS.

Fig. 2*A* illustrates the crossing probabilities, $P_A(\lambda|\lambda_A)$, of these simulations. This is the probability that the system's order parameter function $\lambda(x)$ reaches the order parameter value

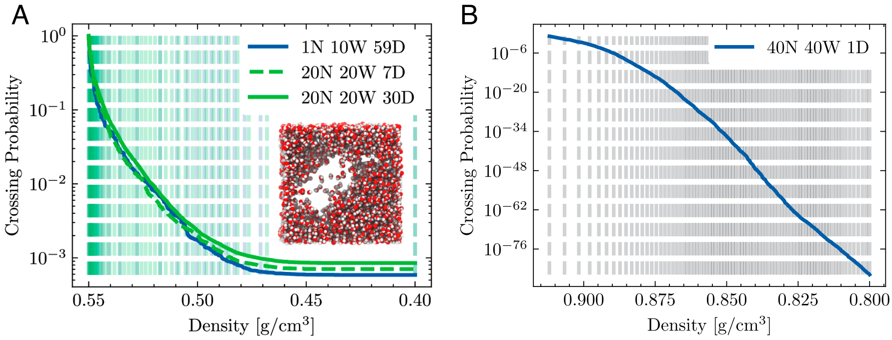


Fig. 2. The crossing probability for density reduction, with the snapshot in (*A*) describing the system at 0.40 g/cm^3. The legend acronyms represent (N)odes, (W)orkers, and (D)ays. The vertical dashed lines indicate the interface placements. (*A*) Metastable liquid water at 573.15 K. Two ∞RETIS simulations are performed on identical, GPU-equipped nodes, where the single node simulation utilizes MPS (*Optimizing GPU Utilization* and Fig. 5) while the multi-node simulation does not. The 30-d plot is the continuation of the 7-d plot. (*B*) The results of a 1-d ∞RETIS simulation of liquid water at the SPC/E boiling temperature equal to 396.0 K based on 40 workers each utilizing their own individual nodes. Based on equilibrium MD runs, an average density of 0.57 g/cm^3 is obtained for *A* and 0.92 g/cm^3 for (*B*).

$\lambda > \lambda_A$ after crossing λ_A in the positive direction without recrossing λ_A (6). In our case, the order parameter was defined as minus the density of the system (the minus signs are omitted in Fig. 2). Hence, the computed probability is reflecting the likelihood of a small density fluctuation below the metastable density of 0.55 g/cm^3 causing the density to continue decreasing until reaching the point of no return. At this point, the system transitions to the gas phase with minimal probability of returning to the metastable liquid state, which occurs when the density falls below 0.45 g/cm^3. This trend is evident as the crossing probability converges to a consistent horizontal plateau beyond this density threshold.

Simulation (ii) requires just 7 d to generate the same total MD steps as simulation (i) would in 59 d. Additionally, from the graph, it is evident that simulation (ii) achieves excellent convergence, with minimal differences in the crossing probability observed after continuing the 7-d simulation up to 30 d. However, simulation (i) benefits from MPS utility, enabling it to run with 10 workers in parallel on a single GPU, maximizing output per node.

To explore the transformative potential of our methodology, we also assessed ∞RETIS's capability in investigating the liquid-to-vapor transition at the actual SPC/E boiling temperature of 396.0 K (23). At the phase transition temperature, the critical nucleus size for the vapor bubble diverges, leading to a vanishingly small rate in the thermodynamic limit. We therefore examined exceedingly rare density fluctuations that do not yet indicate an irreversible transition to the vapor phase, akin to a point of no return. Although these fluctuations are likely to be dependent on system size, their occurrence rate presents an exceptional computational challenge. In this work, we have used this as a litmus test for ∞RETIS, probing its ability to converge calculations of exceedingly small probabilities within a short wall time period when operating on a massively parallel GPU computer.

By employing 40 workers on 40 individual nodes and 80 interfaces for 1 d, ∞RETIS manages to compute the crossing probability and the corresponding rate for the scenario in which random fluctuations lead the system to reach a density below 80% of the stable liquid phase, see Fig. 2*B* and Table 1. To the best of our knowledge, the final value of the crossing probability, astonishingly low in the order of 10^{-86}, represents a world record for the lowest computed crossing probability in a realistic molecular system.

Application II: Chignolin Unfolding. The CLN025 mutant of chignolin is a popular test system for rare event methods, and we examine the unfolding of this mini protein with two different order parameters. The first-order parameter is the RMSD of the protein backbone from the folded state. During this simulation, the system quickly began exploring a set of long-lived misfolded states.

Paths going through these misfolded states were characterized by low weights in their corresponding path ensembles due to their length. In addition, an actual experiment will hardly be able to discriminate between the misfolded and native states, and grouping them into an ensemble of folded structures is more meaningful (28). Based on these arguments, we include the misfolded structures in the folded state definitions and perform an additional simulation with a second-order parameter; a neural network trained on a diffusion map created from a couple of reactive trajectories from previous simulation data, an approach we denote Deep-DM. In Fig. 3*A*, we illustrate the conditional free energy from the first simulation mapped onto the two deep-DM coordinates, in which the misfolded states are apparent.

Fig. 3*A* sheds light on the vast conformational landscape that even a mini protein like chignolin covers during its transition to the unfolded state and illustrates the extensive sampling enabled by our rare event protocol. Notably, we observe a total of 1,000 and 1,600 reactive trajectories during the RMSD and deep-DM simulations, respectively, which allows us to sample a wide range of transition paths. In comparison, brute force Anton simulations of almost twice the length observed around 30 to 40 transitions (24), and we observed 14 transitions in the course of an 80 µs equilibrium simulation.

We also see that instead of a single unfolding route, the protein explores a variety of configurations during the unfolding process, which is not characterized by a single well-defined free energy barrier. This gives rise to the complex kinetic behavior reported previously in long unbiased simulations (24). Such processes can be challenging to model with other methods that rely on assumptions regarding the transition state and the reaction coordinate, which in our protocol can be obtained post-simulation (29).

It is interesting to note the resemblance of the D state to an α-helical structure even though chignolin is a β-hairpin in its native state. Experimental evidence suggests that β-hairpin formation may occur competitively with α-helical formation (30). Fig. 3*C* presents the running averages of the rates for both

Table 1. The results and setup for all the simulations ran in this paper

Simulation	Rate	Flux	$P_A(\lambda_B\|\lambda_A)$	Nodes	Workers	Days
Boiling$_{573.15K}$	4.73e+06 s^{-1} ± 102%	8.05e+09 s^{-1} ± 71%	5.88e−04 ± 45%	1	10	59
Boiling$_{573.15K}$	4.76e+06 s^{-1} ± 54%	5.59e+09 s^{-1} ± 52%	8.51e−04 ± 14%	20	20	30
Boiling$_{396.0K}$	4.03e−75 s^{-1} ± 61%	5.33e+10 s^{-1} ± 13%	7.56e−86 ± 64%	40	40	1
Chignolin$_{RMSD}$	0.17 µs^{-1}± 59%	14,000 µs^{-1}± 19%	1.2e−05 ± 50%	1	16	19
Chignolin$_{Deep-DM}$	0.17 µs^{-1}± 29%	17,000 µs^{-1}± 6 %	9.9e−06 ± 32%	1	10	15
Eq. sim.	0.18 µs^{-1}					
Anton (24)	0.45 µs^{-1}					
Deep-TICA (27)	0.31 µs^{-1}					
HLDA (27)	0.16 µs^{-1}					
Dissociation*	1.17e−01 s^{-1} ± 92%	2.22e+12 s^{-1} ± 13%	5.28e−14 ± 90%	20	40	10
Ref. (25)*	1.29e−01 s^{-1}	2.92e+12 s^{-1}	4.40e−14			

The number of ensembles for each reported simulation is twice the number of workers. *The reported rate, flux, and crossing probability for the dissociation simulation and ref. 25 in this table is for $\lambda_B = 5.0$, which is different from the results reported in ref. 25. Additionally, due to a difference in subcycles (26), the flux and crossing probability between the simulation data and ref. are not directly comparable, but the rate is. Reported errors are estimated based on block-averaging procedures using single SDs.

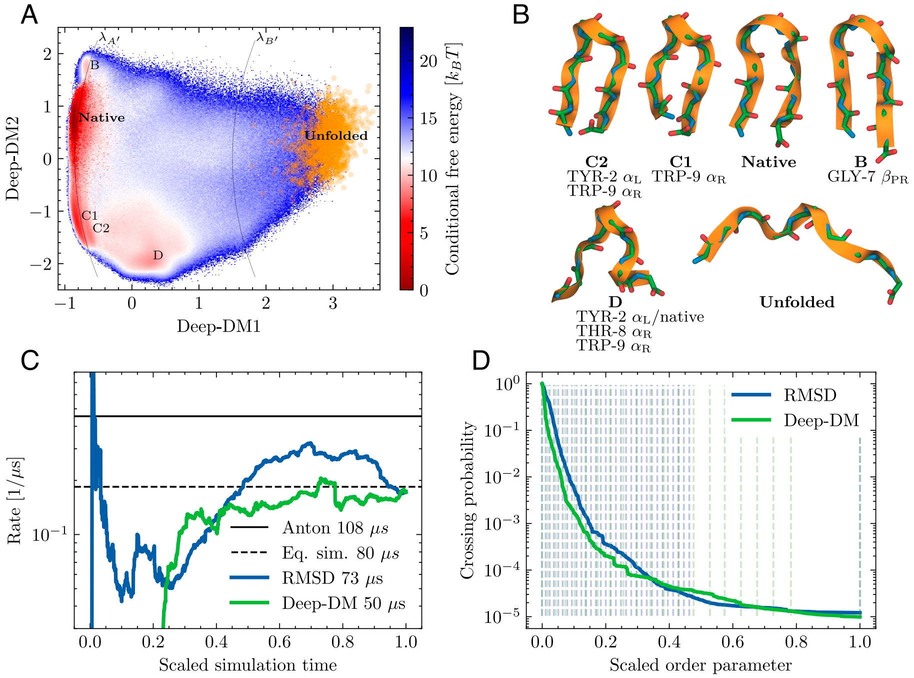


Fig. 3. (*A*) The conditional free energy (i.e., based on phasepoints lying on paths coming from the folded state) mapped onto the deep-DM coordinates, where a set of metastable states is apparent. These results correspond to the simulation with the RMSD order parameter. We do not see a minimum in the path free energy around the unfolded region because of the decreasing probability of reaching such high-order parameter values, given that the path starts in A. We also plot the final phasepoint of each reactive path in orange, which corresponds to a backbone RMSD ≥ 6.0 Å from the folded structure. The curved black lines represent the interface positions of a second set of path simulations with another order parameter, which is a combination of the two deep-DM coordinates. In this second simulation, $\lambda_{A'}$ refers to the folded state interface, which is now a collection of three structures, and $\lambda_{B'}$ refers to the unfolded state. (*B*) Chignolin conformations illustrating the native state, a set of misfolded states (B, C1, C2), a metastable state (D), and a representative sample of the unfolded state that was observed during our simulations. For each metastable state, we also annotate the difference in amino acid conformations compared to the native state. (*C*) The running average of the unfolding rates using the two order parameters, and comparisons with rates obtained from unbiased simulations with Anton (24) and our own equilibrium simulation (Eq. sim.). The legend gives the total simulation time used in the running averages. (*D*) The crossing probabilities from the two simulations and the corresponding interface locations. The order parameter is scaled for comparability.

simulations, while Fig. 3*D* displays the crossing probabilities and interface locations. The calculated rates are in good agreement with those obtained from extensive unbiased simulations and enhanced sampling simulations, even when one of the order parameters incorporates two misfolded states, underscoring the robustness of our approach. A summary of the results can be found in Table 1.

Application III: Ab Initio Water Dissociation. We replicate the RETIS study (25) of calculating the water dissociation reaction rate at 300 K using ab initio MD with the CP2K (31) engine. Satisfactory agreement with the RETIS simulation is obtained from a 10-d ∞RETIS simulation using 40 workers and 80 ensembles, as shown in Fig. 4. A subtle qualitative difference becomes evident within the 3.0 to 4.0 Å shoulder region. The presence of intermediate horizontal plateaus can be attributed to the Grotthuss mechanism, involving a simultaneous double proton transfer (25). This leads to the excess proton residing at an oxygen atom not in direct proximity to the hydroxyl group. Notably, while the original RETIS results suggest that this subprocess consistently reaches completion once initiated, the new ∞RETIS findings paint a more nuanced picture. They reveal a slightly shorter plateau, implying that some double proton transfers may fail and reverse, despite being nearly completed. In terms of wall time, over 1 y was spent running the RETIS simulation, so a rough estimate of the increased wall time efficiency when using ∞RETIS would be $365/10 = 36.5$. A considerable contributor to this difference lies in the sequentiality of the RETIS algorithm. As the average path length generally increases with the ensembles number $[i^+]$, the wall time required to generate new paths increases as well. For this system, trajectories generated by ensembles in the gradual 3.0 to 5.0 Å range can be up to 100 times longer than those from ensembles in the steeper 1.0 to 1.5 Å region (Fig. 4). Consequently, in the context of RETIS, even though the lower ensembles hold the potential for rapid sampling due to their shorter average path lengths, the sequential sampling of higher ensembles leads to extended periods between each ensemble update, as these higher ensembles demand significantly more wall time to generate new trajectories. This sequential challenge is circumvented in the ∞RETIS simulation, where any of the participating workers can initiate MC moves in the lower ensembles whenever they are available. In addition to these algorithmic enhancements, other factors may have contributed to the performance increase, including hardware and software advancements since the original study in 2018.

Optimizing GPU Utilization. Asynchronous replica exchange effectively harnesses the benefits of current and future developments of HPC by allowing the initialization of high worker

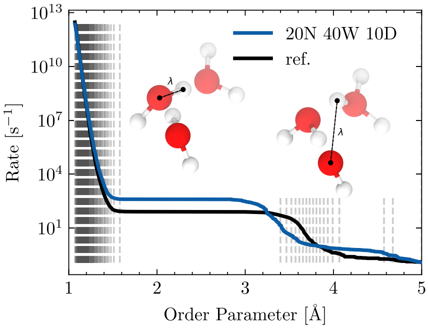


Fig. 4. The rate of water dissociation (crossing probability multiplied by the flux) is compared between the previously reported result, ref. 25, and the result generated from a 10 d ∞RETIS simulation using 20 (N)odes, 40 (W)orkers and 80 ensembles, where each node employs two workers each. The x-axis is the composite order parameter described in ref. 25 and in *Materials and Methods*. While the MD time step is identical for both simulations, there is a slight variation in the time between frames, Δt, caused by differences in the frame-saving rate. Although this discrepancy may influence factors like flux and crossing probability, the product remains independent of it (26), allowing for a direct comparison.

numbers to the high amounts of compute hardware (CPU, GPU, and nodes) available on HPCs. However, an additional benefit is also the effective utilization of NVIDIA MPS (32) when running GPU-accelerated MD, as MPS allows multiple independent processes to concurrently run on the same GPU. For our GPU acceleratable application examples, we observe a 2.4-fold increase in the effective throughput (total ns/day) when running a 12,165 particle boiling system on a node with a 12-core Intel Xeon E5-2690 v3 CPU and an NVIDIA Tesla P100 16 GB GPU, as seen in Fig. 5. With even better scaling, we observe a 6.0-fold increase for a 5,889 particle Chignolin system on a node with a 16-core Intel Xeon E5-2687W CPU and NVIDIA GeForce RTX 3090 GPU. Therefore, large worker numbers (i.e., 10 to 16 in our case) can be readily initialized without necessarily requiring multi-node hardware. MPS on multiple parallel nodes would be even more powerful but was not feasible on the available computing resources.

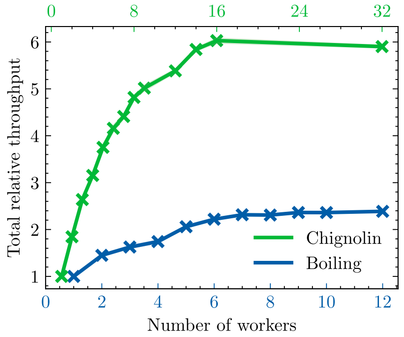


Fig. 5. Total relative throughput (ns/day) for the studied water boiling and chignolin systems running on one GPU-equipped node, with a varying number of concurrently running simulations. At zero parallelization, i.e., when one simulation employs all of the hardware resources of one node, MD throughput averages at 75.0 and 646.2 ns/day. Optimally, with the use of NVIDIA's MPS service, throughputs of 178.8 ns/day (14.9 ns/day × 12 parallels) and 3896.0 ns/day (243.5 ns/day × 16 parallels) are achieved. The irregular spacing between the data points is due to the sharing of CPU cores (constant) between the number of workers (variable). The hardware specification for each system is detailed in *Materials and Methods*. The data points are averages based on 10 repeated trials per data point, with a low to insignificant SD.

Discussion

Utilizing the power of recent path sampling innovations, we have developed an efficient path sampling protocol referred to as ∞RETIS. In challenging realistic applications, our protocol demonstrated outstanding scalability across diverse GPU and CPU computing platforms using both classical and Ab Initio dynamics. Its remarkable sampling efficiency enabled swift convergence of transition rates within high-dimensional systems that previously would require months to years for convergence. With the ∞RETIS algorithm deployed on potent HPC systems, they now succumb within mere days or weeks. This is a significant advancement, as path sampling offers a distinct advantage over other rare event techniques, such as metadynamics (33) and steered MD (34), by enabling the study of completely unbiased dynamics. However, its computational costs have slowed down the widespread adoption of quantitative path sampling simulations in large molecular systems. The algorithmic innovations detailed in this paper are poised to revolutionize this landscape, making previously unattainable systems accessible and potentially guiding experimental discoveries.

Materials and Methods

Rate Calculation. Rates were computed from the RETIS ensembles by writing $k_{AB} = f_A P_A(\lambda_B|\lambda_A)$ where f_A is the frequency for the system to exit state A, and $P_A(\lambda_B|\lambda_A)$ is the crossing probability, the very small chance that after an exit, the system manages to reach state B without revisiting state A. In RETIS, the flux is determined from the average path lengths in the $[0^-]$ and $[0^+]$ path ensembles. The total crossing probability is obtained from the product of local crossing probabilities, $P_A(\lambda_B|\lambda_A) = \prod_{i=0}^{n-1} P_A(\lambda_{i+1}|\lambda_i)$ where $P_A(\lambda_{i+1}|\lambda_i)$ is estimated from the fraction of sampled paths in the $[i^+]$ ensemble that cross the next interface λ_{i+1}. Further improvement in the statistical analysis (29, 35) has been obtained using the weighted histogram analysis method (WHAM) (36). All estimated errors on computed properties have been based on a block-averaging procedure. Additional properties like the conditional free energy (Fig. 3*A*) were obtained using a WHAM (36) reweighting procedure on the collective phase points of the trajectories of all RETIS ensembles (29, 35).

Initialization. Like in standard RETIS and TIS, the interface positions in ∞RETIS are initially configured so that $P_A(\lambda_{i+1}|\lambda_i)$ is approximately the same across different i values, a tuning process conducted during preliminary initialization runs. However, ∞RETIS does not aim for a specific target value like the rule of thumb value of 0.2 (37), which fixes the number of required interfaces. Instead, the number of interfaces $(n+1)$ is based on the available hardware, i.e., the number of workers that can be launched. To ensure plenty availability of free ensembles with sufficient overlap at each infinite swapping step, n is set to be twice the number of workers. Once n is fixed, we aim to place the interfaces such that $P_A(\lambda_{i+1}|\lambda_i) \approx P_A(\lambda_B|\lambda_A)^{(1/n)}$ for all i using an estimation for $P_A(\lambda_B|\lambda_A)$ from the short initialization run.

Sampling. The WF move was employed in all $[i^+]$ ensemble simulations for $0 < i < n$. We determined the parameters N_s and λ_{cap} without conducting an extensive optimization analysis; instead, we chose values that appeared reasonable. This approach led us to use the same N_s value for all ensembles, rather than aiming for ensemble-specific values based on the ratio of each ensemble's average path length to the average path length of the subtrajectories (8). While significantly enhancing the efficiency compared to standard shooting, more efficient parameter sets likely exist. We plan to explore automatic parameter adjustment and initialization in the future. The $[0^-]$ and $[0^+]$ ensembles employed normal shooting without high acceptance. In these ensembles, where the path length of subtrajectories matches that of full paths, the WF move has a reduced impact. Furthermore, the absence of high acceptance implies that the MD-intensive point exchange move $[0^-] \leftrightarrow [0^+]$ can always be accepted. Instead of high acceptance, these ensembles use an early rejection scheme (6) that allows for the interruption and rejection of the generation of excessively long paths, which would have been rejected anyway in the Metropolis–Hastings step.

In quantitative terms, the acceptance of the WF in the boiling simulation at 396.0 K reached 100% due to the absence of a stable *B* state attainable from *A*. Likewise, in the water dissociation study, the WF move demonstrated a similarly remarkable acceptance rate of 98.9%. However, in boiling simulations conducted at the higher temperature of 573.15 K, the WF move exhibited lower acceptances of 74.1% and 84.9% for the simulations utilizing 10 workers (21 interfaces) and 20 workers (41 interfaces) respectively. Rejections predominantly occurred within the last set of easily converging path ensembles, where the λ_i interface required for crossing is already proximate to state *B*. Focusing on the challenging part wherein the system ascends in free energy, disregarding the latter path ensembles where paths have over a 50% likelihood of reaching state *B*, the WF acceptance escalates to, respectively, 89.3% and 94.2%. This hints at the potential for even greater efficiency by adhering to a slightly modified protocol than the one described in the previous section, aligning the $(n-1)$th interface such that $P_A(\lambda_n|\lambda_{n-1}) < 0.5$.

The protein unfolding study exhibited a similar trend in the acceptance rate of the WF move, demonstrating nearly 100% acceptance in the initial path ensembles before decreasing for interfaces closer to state *B*. Across the simulations shown in Fig. 3, the overall WF acceptance rates were 73.8% (RMSD) and 45.3% (Deep-DM). The relatively lower acceptance observed in the latter case is attributed not only to the suboptimal positioning of λ_{n-1}, but also arises from the asymmetric shape of the free energy landscape, requiring λ_{cap} to be placed farther from state *B*, closer to the peak of the barrier. With the current interfaces, the generation of a single $A \rightarrow B$ trajectory tends to provide shooting points predominantly within the basin of attraction of state *B*, which can lead to dramatically low shooting acceptance (38). Consequently, there is a high likelihood that all N_S subtrajectories become unsuccessful. This shows that the very high acceptance and optimal efficiency is achieved with fine tuning of the method's parameters, but even with suboptimal WF parameters, both acceptance and decorrelation are still superior to those achieved with standard shooting especially for the asymmetric barrier case (38).

Code Implementation and Availability. We run an in-house ∞RETIS Python code which mainly consists of PyRETIS (17) function imports together with the use of the Dask (39) package which handles scheduling worker tasks. To start a simulation, the user determines the number of workers to be employed based on the hardware available and the type of system to be simulated. An additional user variable is subcycles, which controls the number of frames to be saved between the number of generated MD steps. For instance, if a trajectory comprises 200,000 MD steps, the trajectory in the path ensemble is delineated by 200 time slices, each corresponding to every 1,000th MD frame when subcycles is set to 1,000. Once the setup is completed, the Python code schedules available workers to perform MD-based MC moves. The running of MD, engine input/output, and data storage are mainly handled by the PyRETIS functions that externally start and stop GROMACS/CP2K simulations. The code used to generate the paper data is available at https://doi.org/10.5281/zenodo.8380343, but an updated code that is under development is accessible via GitHub https://github.com/infretis/.

Simulation Details on Superheated Water Boiling. Superheated liquid water in the form of 4055 H2O water molecules is simulated with periodic boundary conditions and a timestep of 0.5 fs in the NPT ensemble at 1 bar and the two temperatures 396.0 and 573.15 K using Gromacs 2021.5 (40). The temperature and pressure are kept constant by applying a V-rescale thermostat (41) of 2.5 ps relaxation time and a C-rescale barostat (42) with a relaxation time of 10 ps. As with the previous TPS studies (20, 21), the SPC/E water model (43) is also used. The order parameter is simply the water density, and the initial reactive trajectories are obtained by quickly heating an equilibrated system. The ∞RETIS simulation ran with a subcycle of 1,000 and the number of WF subtrajectories equals 4. The simulations were run on HPC nodes consisting of 12 core Intel Xeon E5-2690 v3 CPU and NVIDIA Tesla P100 16 GB GPU.

Simulation Details on Chignolin Unfolding. The simulations of the CLN025 mutant of chignolin are performed with the setup described by Bonatti (44), using their provided input files available online. The mini protein is modeled with the CHARMM22* force field and solvated with TIP3P water molecules at 340 K. The terminal amino acids and the ASP and GLU amino acids are modeled in their charged states, and the system is neutralized by adding two sodium ions. The equations of motion are integrated with a timestep of 2 fs using the velocity Verlet scheme, and canonical sampling is achieved with the V-rescale thermostat (41).

The first path sampling simulation is performed with an order parameter defined as the RMSD between the protein backbone and the folded structure (the average structure from a long simulation, not the crystal structure). The folded and unfolded interfaces were given by an RMSD of 0.6 and 6.0 Å, respectively, and we used an interface cap at 4.0 Å. Multiple misfolded states were observed during this simulation. We train a neural network that includes the B and C1 states as part of the folded ensemble in the following manner; we first construct a diffusion map with the approach outlined in ref. 45 using chignolin configurations representing the native folded state, the misfolded states observed in the first path simulations, unfolded configurations, and a set of configurations from the transition path ensemble. The protein backbone RMSD is used as a distance metric to construct the diffusion map. We then train the neural network directly on the two leading eigenvectors of the diffusion map. We use the 741 interatomic distances between backbone atoms as input features for the network, with architecture 741-50(ReLu)-25(ReLu)-12(ReLu)-2. This is motivated and similar in spirit to the approach described in ref. 46, except that we do not fit the network to pre-assigned positions, but rather on the output of the diffusion map. Using this approach, we can discriminate between the folded, misfolded, and unfolded states as well as some other metastable states. The A and B interfaces were given by an order parameter of −0.85 and 1.5, respectively, and we used an interface cap of 1.2.

For all systems, we use 16 Intel Xeon E5-2687W CPUs and partition 1 NVIDIA447 GeForce RTX 3090 GPU among the workers using MPS. For the RMSD, we use 16 workers, and for the deep-DM system, we use 10 workers. The ∞RETIS timestep was 4 ps (2000 MD integration steps with a 2 fs timestep), and the number of WF subtrajectories was 3.

Simulation Details on Ab Initio Water Dissociation. We replicate the previous RETIS study (25), i.e., 32 water molecules are simulated in a periodic 9.85 Å cubic box with ab initio DFT MD using CP2K 9.1 (31). The DFT calculations use Becke–Lee–Yang–Parr functional (47, 48) with a DZVP-MOLOPT (49) basis set and a plane-wave cutoff of 280 Ry. The MD simulations were run with a timestep of 0.5 fs, the number of subcycles is set to 5 for the ∞RETIS simulation and the number of WF subtrajectories is set to 2. The new velocities generated by the shooting/WF move are drawn from a Maxwell–Boltzmann distribution corresponding to an average temperature of 300 K. The order parameter is the longest O-H distance in the case where no dissociated species exist in the system. When $OH^- + H_3O^+$ pair(s) are detected, the order parameter becomes the shortest distance between the oxygen in OH^- and hydrogens in H_3O^+. The ∞RETIS simulation was run on the Sigma2 HPC system Saga with 20 nodes equipped with Intel Xeon-Gold 6138 CPUs, where two workers were run on each node.

Data, Materials, and Software Availability. All study data are included in the main text.

ACKNOWLEDGMENTS. We thank Zhiliang Zhang for the helpful comments on improving our paper. We acknowledge funding support from the Research Council of Norway (Grant No. 275506). We also thank Sigma2, the National Infrastructure for high-performance computing (HPC) and Data Storage in Norway and the HPC infrastructure IDUN at the NTNU.

Author affiliations: [a]Department of Chemistry, Norwegian University of Science and Technology, Trondheim N-7491, Norway; and [b]Department of Chemistry, Utrecht University, Utrecht 3584 CH, Netherlands

1. L. Grajciar *et al*., Towards operando computational modeling in heterogeneous catalysis. *Chem. Soc. Rev.* **47**, 8307–8348 (2018).
2. D. W. Borhani, D. E. Shaw, The future of molecular dynamics simulations in drug discovery. *J. Comput. Aided Mol. Des.* **26**, 15–26 (2012).
3. K. Shmilovich *et al*., Discovery of self-assembling π-conjugated peptides by active learning-directed coarse-grained molecular simulation. *J. Phys. Chem. B* **124**, 3873–3891 (2020).
4. B. Peters, *Reaction Rate Theory and Rare Events* (Elsevier, 2017).
5. P. G. Bolhuis, D. Chandler, C. Dellago, P. L. Geissler, Transition path sampling: Throwing ropes over rough mountain passes, in the dark. *Ann. Rev. Phys. Chem.* **53**, 291–318 (2002).
6. T. S. van Erp, D. Moroni, P. G. Bolhuis, A novel path sampling method for the calculation of rate constants. *J. Chem. Phys.* **118**, 7762–7774 (2003).
7. T. van Erp, Reaction rate calculation by parallel path swapping. *Phys. Rev. Lett.* **98**, 268301 (2007).
8. D. T. Zhang, E. Riccardi, T. S. van Erp, Enhanced path sampling using subtrajectory Monte Carlo moves. *J. Chem. Phys.* **158**, 024113 (2023).
9. S. Roet, D. T. Zhang, T. S. van Erp, Exchanging replicas with unequal cost, infinitely and permanently. *J. Phys. Chem. A* **126**, 8878–8886 (2022).
10. N. Plattner *et al*., An infinite swapping approach to the rare-event sampling problem. *J. Chem. Phys.* **135**, 134111 (2011).
11. L. R. Pratt, A statistical method for identifying transition states in high dimensional problems. *J. Chem. Phys.* **85**, 5045 (1986).
12. C. Dellago, P. G. Bolhuis, F. S. Csajka, D. Chandler, Transition path sampling and the calculation of rate constants. *J. Chem. Phys.* **108**, 1964 (1998).
13. E. Riccardi, O. Dahlen, T. S. van Erp, Fast decorrelating Monte Carlo moves for efficient path sampling. *J. Phys. Chem. Lett.* **8**, 4456–4460 (2017).
14. N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, E. Teller, Equation of state calculations by fast computing machines. *J. Chem. Phys.* **21**, 1087–1092 (1953).
15. W. Hastings, Monte-Carlo sampling methods using Markov chains and their applications. *Biometrika* **57**, 97–109 (1970).
16. D. W. H. Swenson *et al*., A Python framework for path sampling simulations. 2. Building and customizing path ensembles and sample schemes. *J. Chem. Theory Comput.* **15**, 837–856 (2019).
17. E. Riccardi, A. Lervik, S. Roet, O. Aarøen, T. S. van Erp, PyRETIS 2: An improbability drive for rare events. *J. Comput. Chem.* **41**, 370–377 (2020).
18. D. G. Glynn, The permanent of a square matrix. *Eur. J. Comb.* **31**, 1887–1891 (2010).
19. P. Lundow, K. Markström, Efficient computation of permanents, with applications to boson sampling and random matrices. *J. Comput. Phys.* **455**, 110990 (2022).
20. D. Zahn, How does water boil? *Phys. Rev. Lett.* **93**, 227801 (2004).
21. K. Karalis, D. Zahn, N. I. Prasianakis, B. Niceno, S. V. Churakov, Deciphering the molecular mechanism of water boiling at heterogeneous interfaces. *Sci. Rep.* **11**, 19858 (2021).
22. R. E. Apfel, Water superheated to 279.5 °C at atmospheric pressure. *Nat. Phys. Sci.* **238**, 63–64 (1972).
23. M. Fugel, V. C. Weiss, A corresponding-states analysis of the liquid-vapor equilibrium properties of common water models. *J. Chem. Phys.* **146**, 064505 (2017).
24. K. Lindorff-Larsen, S. Piana, R. O. Dror, D. E. Shaw, How fast-folding proteins fold. *Science* **334**, 517–520 (2011).
25. M. Moqadam *et al*., Local initiation conditions for water autoionization. *Proc. Natl. Acad. Sci. U.S.A.* **115**, E4569–E4576 (2018).
26. P. G. Bolhuis, D. W. H. Swenson, Transition path sampling as Markov Chain Monte Carlo of trajectories: Recent algorithms, software, applications, and future outlook. *Adv. Theory Simul.* **4**, 2000237 (2021).
27. D. Ray, N. Ansari, V. Rizzi, M. Invernizzi, M. Parrinello, Rare event kinetics from adaptive bias enhanced sampling. *J. Chem. Theory Comput.* **18**, 6500–6509 (2022).
28. P. Kührová, A. De Simone, M. Otyepka, R. B. Best, Force-field dependence of chignolin folding and misfolding: Comparison with experiment and redesign. *Biophys. J.* **102**, 1897–1906 (2012).
29. T. S. van Erp, M. Moqadam, E. Riccardi, A. Lervik, Analyzing complex reaction mechanisms using path sampling. *J. Chem. Theory Comput.* **12**, 5398–5410 (2016).
30. C. M. Davis, S. Xiao, D. P. Raleigh, R. B. Dyer, Raising the speed limit for β-hairpin formation. *J. Am. Chem. Soc.* **134**, 14476–14482 (2012).
31. T. D. Kühne *et al*., CP2K: An electronic structure and molecular dynamics software package - Quickstep: Efficient and accurate electronic structure calculations. *J. Chem. Phys.* **152**, 194103 (2020).
32. Maximizing GROMACS Throughput with Multiple Simulations per GPU Using MPS and MIG (2021). Accessed 25 September 2023.
33. A. Laio, M. Parrinello, Escaping free-energy minima. *Proc. Natl. Acad. Sci. U.S.A.* **99**, 12562 (2002).
34. S. Izrailev *et al*., "Steered molecular dynamics" in *Computational Molecular Dynamics: Challenges, Methods, Ideas*, P. Deuflhard, Ed. (Springer, Heidelberg, 1999), pp. 39–65.
35. J. Rogal, W. Lechner, J. Juraszek, B. Ensing, P. G. Bolhuis, The reweighted path ensemble. *JCP* **133**, 174109 (2010).
36. A. Ferrenberg, R. Swendsen, Optimized Monte-Carlo data-analysis. *Phys. Rev. Lett.* **63**, 1195–1198 (1989).
37. R. Cabriolu, K. M. S. Refsnes, P. G. Bolhuis, T. S. van Erp, Foundations and latest advances in replica exchange transition interface sampling. *J. Chem. Phys.* **147**, 152722 (2017).
38. Z. F. Brotzakis, P. G. Bolhuis, A one-way shooting algorithm for transition path sampling of asymmetric barriers. *J. Chem. Phys.* **145**, 164112 (2016).
39. Dask Development Team, Dask: Library for dynamic task scheduling (2016).
40. M. J. Abraham *et al*., GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX* **1–2**, 19–25 (2015).
41. G. Bussi, D. Donadio, M. Parrinello, Canonical sampling through velocity rescaling. *J. Chem. Phys.* **126**, 014101 (2007).
42. M. Bernetti, G. Bussi, Pressure control using stochastic cell rescaling. *J. Chem. Phys.* **153** (2020).
43. H. J. C. Berendsen, J. R. Grigera, T. P. Straatsma, The missing term in effective pair potentials. *J. Phys. Chem.* **91**, 6269–6271 (1987).
44. L. Bonati, G. Piccini, M. Parrinello, Deep learning the slow modes for rare events sampling. *Proc. Natl. Acad. Sci. U.S.A.* **118**, e2113533118 (2021).
45. S. B. Kim, C. J. Dsilva, I. G. Kevrekidis, P. G. Debenedetti, Systematic characterization of protein folding pathways using diffusion maps: Application to Trp-cage miniprotein. *J. Chem. Phys.* **142** (2015).
46. D. Ray, E. Trizio, M. Parrinello, Deep learning collective variables from transition path ensemble. *J. Chem. Phys.* **158**, 204102 (2023).
47. A. D. Becke, Density-functional exchange-energy approximation with correct asymptotic behavior. *Phys. Rev. A* **38**, 3098–3100 (1988).
48. C. Lee, W. Yang, R. G. Parr, Development of the Colle-Salvetti correlation-energy formula into a functional of the electron density. *Phys. Rev. B* **37**, 785–789 (1988).
49. J. VandeVondele, J. Hutter, Gaussian basis sets for accurate calculations on molecular systems in gas and condensed phases. *J. Chem. Phys.* **127**, 114105 (2007).

ISBN 978-82-326-8022-1 (printed ver.)
ISBN 978-82-326-8021-4 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

NTNU
Norwegian University of
Science and Technology